Probabilistic Models for Demand Supply Prediction in The Eenergy Sector

Muluken Regas Eressa

Université Gustave Eiffel

Probabilistic Models for Demand Supply Prediction in the Energy Sector

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale: Mathématiques et Sciences et Technologies de l'Informations et de la Communication (MSTIC)
Spécialité de doctorat: Informatique
Unité de recherche : Laboratoire d'Informatique Gaspard-Monge (LIGM)

Thèse présentée et soutenue à l'Université Gustave Eiffel, le 26/01/2024, par :

Muluken Regas ERESSA

Composition du Jury

Ye-Qiong SONG Professor at University of Lorraine (ex-INPL) Université de Lorraine - ENSEM	Président du jury
Nadjib AIT SAADI Professor at UVSQ Paris-Saclay University	Examinateur
Samia BOUZEFRANE Professeure des universités, Laboratoire CEDRIC Conservatoire National des Arts et Métiers	Rapportrice
Hakima CHAOUCHI Professor at Telecom SudParis Institut Mines Telecom, Telecom Sud Paris	Rapportrice

Encadrement de la thèse

Rami LANGAR Gustave Eiffel University, LIGM (UMR 8049)	Directeur de thèse
Hakim BADIS Gustave Eiffel University, LIGM (UMR 8049)	Co-Directrice de thèse
Dorian GROSSO METRON - Energy Intelligence for Industries	Co-Encadrant de thèse

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at the Gustave Eiffel Université, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications. I have read the University's current research ethics guidelines, and accept responsibility for the conduct of the procedures in accordance with the University's rules and regulations.

Acknowledgment

First and foremost, I would like to thank my creator for giving me the opportunity to live, learn and expand my understanding. Secondly, I would like to take this opportunity to thank Professor Laurent George, who selected me for this position, and who is no longer with us. He was a man of honor, dedication and a good mentor. May the almighty give him eternal peace and rest. I would also like to thank my supervisors Hakim, Rami, Dorian, who took the responsibility, extended a helping hand and provided the needed guidance without reservation on this PhD journey. Finally, I would like to show my appreciation to all the staff of METRON and UGE, my friends and families for the care, compassion and encouragement.

"Citation"

Abstract

A great technological strides have been carried out in the past to enhance the efficiency of production and consumption of energy. The topological change in electric grid from centralized to smart networks is one of them. The traditional grid consists of a centralized large scale generators that are stationed far from the consumption site. Such arrangement helps the utilities to generate power for large consumer bases at a time. In doing so, it nullify the inconsistency in individual energy consumption and flattens the average energy requirement. Inherently, this improves the predictability of the demand. As such, load anticipation and optimal generator dispatch will be easier. In contrary, smart grid relies on smaller and decentralized generation units. The majority of these units run on renewable energy resources and are located closer to the consumer sides. Being smaller in size increases the units affordability which could potentially turn consumers into intermittent energy suppliers. Consequently, the previous uni-directional communication/energy flow from producers to consumers is now changed into bi-directional. These has generated a huge volume of data. Since it is composed of a smaller distributed units, the convenience of bulk power estimation is no longer valid. As a result, there is high variability both in energy production and consumption. Hence, a successful demand response scheme needs better predictive models at the either ends of the power flow. These predictive models should address the variability of supply and demand, handle the big data generated and a provide a marginal profit gain for all parties involved in production, transmission and consumption of electricity. This is where we believe probabilistic models will play a vital role for a successful implementation of demand response optimization.

Probabilistic models can anticipate the uncertainties that could arise during system interactions. These uncertainties can emanate from the very system we are trying to control or can be external to the system. Either ways, in the presence of these uncertainties we can not completely be sure if our estimation is correct. We can take our recent covid pandemic as a good example as to how environmental factors affect the accuracy of a predictive model. The likelihood of this event was closer to none. And yet, despite their parametric nature, most of the predictive models utilized by the energy companies in Europe couldn't anticipate the sudden drop in electricity demand. Miscalculation will be even more damaging to a cascaded system where the decision of the later sub-system is dependent on the output of the former. Such arrangement is prone to error compounding. If the system manipulate deterministic predictive models for decision making, such error augmentation will have a profound effect on its accuracy.

In contrary, a probabilistic model considers all variables and system interactions as random variables that are bound to vary. Therefore, it offers a full spectrum for a possible system outputs taking into account the inputs randomness. As such, it provides a framework for uncertainty propagation so that all subsequent subsystems work on a probable outcomes rather than a singular value which may or might not be the true value. This enables the system to make an informed decision corresponding to the anticipated uncertainty. Thereby making it more robust and reliable. Here, we would like to point out that we are not making an objective assessment on the quality of a predictive model. We are merely considering a possible approach that could improve the accuracy of estimation, as it is dependent on many factors. For example, the availability and quality of data, the modeling approach, forecast horizon, hidden associations between predictors, sudden environmental changes, socioeconomic and political dynamics are some these factors that can impacts predictive accuracy. However, at the bare minimum a good predictive model should exhibit a minimized prediction interval (PI), maximized coverage probability and a robust response to uncertainty.

Quantifying uncertainty using interval width and coverage probability as quality met-

rics and combining it with accuracy enhances the model validity for decision making. Forecast models, be it, parametric, semi-parametric or non-parametric, have been studied to address and improve these qualities. Parametric models are simple, elegant and interpretable. However, because of their constrained parameter space, they are also highly influenced by unmodeled dynamics. On the other hand, non-parametric models have unconstrained parameter space. Consequently, they have the ability to change the function space in order to fit the given observation. These parametric models and their variants have been used by Energy companies for demand-supply prediction. And, to cope up the current trend in technological advances, they are still investing a lot on a new predictive algorithms. Though, the acceptable degree of compromise with regard to model complexity, reliability, accuracy and computational resource requirement remain subjected to the needs of the company and the specific problem at hand.

This thesis investigate probabilistic predictive models based on the Gaussian process and deep learning for electricity demand forecasting. As Gaussian processes are kernel-based predictive models, their performance is constrained by the type, number and dimension of the selected kernel. To address these limitations, first it proposes a new gaussian approximation technique that address the Bayesian computational bottleneck. Second, it proposes a stochastic compositional kernel estimation algorithm using the proposed gaussian approximation as the underlying model. Third, it follows an iterative procedure using cross-validation for selecting an optimal combinations of kernels that best explain the data generating model. Furthermore, it also tries to address the limitation of maximum likelihood approach which is usually employed in probabilistic deep learning models and yet fails in guaranteeing a minimized interval width and maximised coverage probability for the forecasted points. This thesis proposes a new training algorithm for neural networks. The proposed distribution based lower upper bound estimation algorithm encompasses interval width and coverage probability as quality metrics with adaptive parameters that guarantee the needed performance compared to other alternative techniques.

The suggested approaches enhance the deployment of Gaussian and deep learning models in the energy sector. The bound estimation model for a minimized prediction interval and maximized coverage probability, can help potential energy suppliers in sizing generators which will result in a marginal profit gain. In addition, the kernel estimation algorithm can simplify the application of kernel-based learning to those who find kernel selection vague. To the experienced, it can give a preliminary insight into the structure of the kernels that could potentially fit the data. The randomized column sampling technique could offer an alternative method for a fast Gaussian model building and approximation that is scalable to large data. Furthermore, the bound estimation, in addition to providing a forecast distribution to a point estimate neural models, it can also serve as a good starting point to an alternative probabilistic model training in deep neural nets.

Résumé

D'énormes progrès technologiques ont été réalisés dans le passé pour améliorer l'efficacité de la production et de la consommation d'énergie. Le changement topologique dans le réseau électrique, passant d'une structure centralisée à des réseaux intelligents, en fait partie. Le réseau traditionnel se compose de générateurs à grande échelle centralisés, situés loin du site de consommation. Cette disposition aide les services publics à produire de l'électricité pour de grandes bases de consommateurs simultanément. Ce faisant, cela annule les incohérences dans la consommation individuelle d'énergie et aplatit le besoin énergétique moyen. Par nature, cela améliore la prévisibilité de la demande. Ainsi, l'anticipation de la charge et la répartition optimale des générateurs seront plus faciles. En revanche, le réseau intelligent repose sur des unités de génération plus petites et décentralisées. La majorité de ces unités fonctionnent avec des ressources énergétiques renouvelables et sont situées plus près des consommateurs. Leur taille plus réduite rend ces unités plus abordables, ce qui pourrait potentiellement transformer les consommateurs en fournisseurs d'énergie intermittente. Par conséquent, la communication et le flux d'énergie unidirectionnels précédents, des producteurs aux consommateurs, sont désormais devenus bidirectionnels. Cela a généré un volume énorme de données. Composé d'unités distribuées plus petites, le confort de l'estimation de puissance en vrac n'est plus valable. En conséquence, il existe une grande variabilité à la fois dans la production et la consommation d'énergie. Par conséquent, un schéma de réponse à la demande réussi nécessite de meilleurs modèles prédictifs aux deux extrémités du flux d'énergie. Ces modèles prédictifs doivent prendre en compte la variabilité de l'offre et de la demande, gérer les grandes données générées et fournir un gain de profit marginal pour toutes les parties impliquées dans la production, la transmission et la consommation d'électricité. C'est là que nous pensons que les modèles probabilistes joueront un rôle crucial dans la mise en œuvre réussie de l'optimisation de la réponse à la demande.

Les modèles probabilistes peuvent anticiper les incertitudes qui pourraient survenir lors des interactions système. Ces incertitudes peuvent émaner du système même que nous essayons de contrôler ou peuvent être externes au système. Dans les deux cas, en présence de ces incertitudes, nous ne pouvons pas être complètement sûrs que notre estimation est correcte. Nous pouvons prendre notre récente pandémie de covid comme un bon exemple de la façon dont les facteurs environnementaux affectent l'exactitude d'un modèle prédictif. La probabilité de cet événement était proche de zéro. Et pourtant, malgré leur nature paramétrique, la plupart des modèles prédictifs utilisés par les entreprises énergétiques en Europe n'ont pas pu anticiper la baisse soudaine de la demande d'électricité. Une erreur de calcul serait encore plus dommageable pour un système en cascade où la décision du sous-système ultérieur dépend de la sortie du précédent. Une telle disposition est sujette à une augmentation des erreurs. Si le système manipule des modèles prédictifs déterministes pour la prise de décision, une telle augmentation des erreurs aura un effet profond sur sa précision.

En revanche, un modèle probabiliste considère toutes les variables et les interactions système comme des variables aléatoires qui sont destinées à varier. Par conséquent, il offre un spectre complet pour des sorties système possibles en tenant compte de l'aléatoire des entrées. En tant que tel, il fournit un cadre pour la propagation de l'incertitude afin que tous les sous-systèmes ultérieurs travaillent sur des résultats probables plutôt que sur une valeur singulière qui peut être ou non la vraie valeur. Cela permet au système de prendre une décision éclairée correspondant à l'incertitude anticipée, le rendant ainsi plus robuste et fiable. Nous tenons à souligner que nous ne faisons pas une évaluation objective de la qualité d'un modèle prédictif. Nous considérons simplement une approche possible qui pourrait améliorer la précision de l'estimation, car elle dépend de nombreux facteurs. Par

exemple, la disponibilité et la qualité des données, l'approche de modélisation, l'horizon de prévision, les associations cachées entre les prédicteurs, les changements environnementaux soudains, les dynamiques socio-économiques et politiques sont quelques-uns de ces facteurs qui peuvent influencer la précision prédictive. Cependant, au minimum, un bon modèle prédictif devrait présenter un intervalle de prédiction minimisé (PI), une probabilité de couverture maximisée et une réponse robuste à l'incertitude.

Quantifier l'incertitude en utilisant la largeur de l'intervalle et la probabilité de couverture comme métriques de qualité et les combiner avec l'exactitude améliore la validité du modèle pour la prise de décision. Les modèles de prévision, qu'ils soient paramétriques, semi-paramétriques ou non paramétriques, ont été étudiés pour aborder et améliorer ces qualités. Les modèles paramétriques sont simples, élégants et interprétables. Cependant, en raison de leur espace paramétrique contraint, ils sont également fortement influencés par des dynamiques non modélisées. En revanche, les modèles non paramétriques ont un espace paramétrique non contraint. Par conséquent, ils ont la capacité de modifier l'espace des fonctions afin de s'adapter à l'observation donnée. Ces modèles paramétriques et leurs variantes ont été utilisés par les entreprises énergétiques pour la prédiction de l'offre et de la demande. Et, pour faire face à la tendance actuelle des avancées technologiques, elles investissent toujours beaucoup dans de nouveaux algorithmes prédictifs. Cependant, le degré acceptable de compromis en ce qui concerne la complexité du modèle, la fiabilité, la précision et les besoins en ressources computationnelles restent soumis aux besoins de l'entreprise et au problème spécifique en question.

Cette thèse examine des modèles prédictifs probabilistes basés sur le processus Gaussien et l'apprentissage profond pour la prévision de la demande d'électricité. Comme les processus Gaussiens sont des modèles prédictifs basés sur les noyaux, leur performance est contrainte par le type, le nombre et la dimension du noyau sélectionné. Pour remédier à ces limitations, premièrement, elle propose une nouvelle technique d'approximation gaussienne qui aborde le goulot d'étranglement computationnel Bayésien. Deuxièmement, elle propose un algorithme d'estimation de noyau compositionnel stochastique utilisant l'approximation gaussienne proposée comme modèle sous-jacent. Troisièmement, elle suit une procédure itérative utilisant la validation croisée pour sélectionner une combinaison optimale de noyaux qui explique au mieux le modèle de génération de données. De plus, elle tente également de résoudre la limitation de l'approche du maximum de vraisemblance, qui est généralement employée dans les modèles d'apprentissage profond probabilistes et qui échoue à garantir une largeur d'intervalle minimisée et une probabilité de couverture maximisée pour les points prévus. Cette thèse propose un nouvel algorithme d'entraînement pour les réseaux neuronaux. L'algorithme de estimation basé sur la distribution proposé englobe la largeur de l'intervalle et la probabilité de couverture comme métriques de qualité avec des paramètres adaptatifs qui garantissent les performances nécessaires par rapport à d'autres techniques alternatives.

Contents

List of Figures

List of Tables

List of Acronyms

GPR	Gaussian Process Regression
GP	Gaussian Process
NN	Neural nets
PICP	Prediction Interval Coverage Probability
MPIW	Mean Prediction Interval Width
RMSE	Root Mean Squared Error
MSE	Mean Squared Error
RSGA	Random Sparse Gaussian Approximation
VGA	Variational Gaussian Approximation
DTC	Deterministic Training Conditionals
FITC	Fully Independent Training Conditionals
MCMC	Markov Chain Monte Carlo
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
BNN	Bayesian Neural Network
CNN	Convolutional Neural Network
LR	Linear Regression
SVR	Support Vector Regression
RFR	Random Forest Regression
GBR	Gradient Boosting Regression
DTR	Decision Tree Regression
SVM	Support Vector Machine
BCD	Bayesian Clustering by Dynamics
MLP	Multi Layer Perceptron
ELBO	Evidence Lower Bound
MA	Moving Average
ARIMA	Auto Regressive Integrated Moving Average
ARIMAX	Auto Regressive Integrated Moving Average with Exogeneous Input
SARIMA	Seasonal Auto Regressive Integrated Moving Average
SARIMAX	Seasonal Auto Regressive Integrated Moving Average with Exogeneous Input
VAR	Vector Auto Regression
BSTS	Bayesian Structural Time Series
BLR	Bayesian Linear Regression
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
SES	Single/Simple Exponential Smoothing
DES	Double Exponential Smoothing
TES	Triple Exponential Smoothing
RBF	Radial Basis Function

Publications

Journals

1. ERESSA Muluken Regas, Hakim Badis, and Rami Langar, “A Comprehensive Review on Scalable Gaussian Processes”, 2023. (Submitted)

International Conferences

1. ERESSA Muluken Regas, Hakim Badis and Dorian Grosso, ”Distribution Based Upper Lower Bound Estimation In Deep Neural Nets” The 21^{st} International Conference on Machine Learning and Application (IEEE ICMLA 2022), Nassau, Bahamas, 2022. (Published)

2. ERESSA Muluken Regas, Hakim Badis, Rami Langar and Dorian Grosso, ”Stochastic Compositional Kernel Estimation for Gaussian Process Models”, The 15^{th} International Conference on Machine Learning and Computing (ICMLC 2023), Zhuhai, China, 2023. (Published)

3. ERESSA Muluken Regas, Hakim Badis, Laurent George and Dorian Grosso, ”Sparse Variational Gaussian Process with Dynamic Kernel for Electricity Demand Forecasting”, 2022 IEEE 7th International Energy Conference (ENERGYCON), Riga, Latvia, 2022. (Published)

4. ERESSA Muluken Regas, Hakim Badis, Rami Langar and Dorian Grosso, ”Random Sparse Approximation for a Fast and Scalable Gaussian Process”, 2022 2nd International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, 2022. (Published)

Motivation

Energy demand has never been a problem for the energy sector in the past. The accessibility and affordability of cheap oil has minimized the volatility in the energy market. Low population growth index, the absence of environmental awareness and small scale industrialisation are also few other factors that has naturally helped in keeping the demand supply equilibrium. As a result, energy security was not considered as a concern and an issue for most countries. However, the rapid change in world economy, urbanization, demography and the sharp rise in the price of oil has ended the era of abundance. Despite the price, the worsening climatic conditions and its environmental risks has made it clear that we need to limit and minimize the utilization of carbon based energy resources.

This trend toward the green economy and zero-carbon society means that the future energy demand will be met by an environmental friendly and yet seasonal sources of energies. The integration of these sources into the grid to meet the daily and future demands requires a change of perspective with regard to our energy policy, grid architecture, energy production, storage, transportation and consumption. This creates a hub of integrated, dynamic and complex systems, all working together to bring the demand supply equilibrium. These systems such as the energy grid, telecommunication, transportation networks and other connected devices come with their own complexity and associated uncertainties. Additionally, the intermittent nature of the energy sources inherently puts an extra layer of uncertainty on the energy demand. Even if it is not up to the expected level, grid modernization and utilization of renewable energy resources is being carried out around the world. But, presently, the majority of the energy demand is still covered by the expensive carbon based energy sources.

At a time when energy is most expensive, the peak energy demand generated due to power plant failures and seasonal shifts hurts the productivity of manufacturing sectors and the profitability of the energy suppliers. As a result, during the period of peak demand, utility companies are adopting the demand response scheme which allows them to reduce the power delivered to the customers instead of buying it at the international market. This mitigation strategy should be implemented in a way that maximise the profitability of customers as well as the utility companies. Consequently, the success of the demand response paradigm, among other things rest on the accuracy of predicting the energy demand and supply before hand. Currently, parametric models are the goto predictive models in the energy sector. Though they are simplistic and efficient, they can't handle complex feature interactions.

Nowadays energy companies are investing heavily on non-parametric predictive models. Predictive models that takes into account the anticipated volatility of the demand and supply could help utility companies in minimizing risk and maximising profit during the decision making process. In that spirit, this thesis aim to investigate the implementation of probabilistic predictive models in the energy sector. Yet, the uncertainties that arise due to the intermittent nature of the energy supply, the interactions of various systems and the huge data that is generated as a result of it, invites serious questions regarding the efficiency of these models. Questions related to accuracy of prediction, computational efficiency, prediction interval width, coverage probability and more impor-

tantly their scalability to large data. To that end, this dissertation aims to address these questions through the objectives stated below

Objectives

This dissertation primarily focus on investigating the feasible approaches for enhancing the predictive accuracy and computational efficiency of Gaussian process and probabilistic deep learning models. As such,

- It investigating the implementation and suitability of non-parametric probabilistic machine and deep learning models. Analyse their predictive efficiency and computational performance. And, propose an alternative approximation to ensure their scalability.
- Propose an efficient kernel search algorithm to rectify the time constraints imposed on model training and evaluation.
- Propose a new algorithm that will allow a predictive distribution for deep learning models.

Main Contributions

Gaussian process is a kernel-based non-parametric predictive model. Its Bayesian framework allowed it to provide a predictive distribution as opposed to point estimates. However, its performance is constrained by the size of the data, the kernel type and the inference framework. For machine learning models, inference and generalization are directly related to the amount and quality of data available. More data presents better opportunity to learn. These fact is especially true for non-parametric models. Specifically models that follow kernel-based learning, their predictive performance is also dependent on the type and number of selected kernels. The Gaussian process (GP) employ these kernels in order to mold a prior distribution on the given data. Consequently, its predictive performance is dependent on the type of kernels selected for pattern discovery within the data. The kernel matrix dimension is another aspect that presents a computational challenge when analyzing big data. The size of the kernel matrix is equivalent to the number of data points considered for fitting. As such, training GP models on big data incurs a huge computational cost and memory requirement. Thereby inadvertently constraining the underlying model. This is the very fact that forbade the scalability of Gaussian process to big data.

The computational hurdles being the main issue, the success of kernel based learning is also dependent on the users ability to select and compose appropriate kernels. Algorithmic based automatic kernel selection, although possible, it is computationally intensive. In addition to that, kernel combinations compounds the number of hyperparameters which directly affects the computational efficiency of the model. As a result, the direct implementation of kernel-based learning on complex data has been challenging. Regardless, various methods have been suggested for its optimal estimation. For example, exhaustive, grid, randomized and non-parametric search methods are few notable mentions. The effectiveness of these approaches is dependent on the intricacies of the data and the frameworks in which they operate. For example, in Gaussian models, the kernel dimension presents a challenge for suitable kernel assessment. In the case of variational and MCMC-based models, the time complexity required for the ELBO and posterior convergence hinders the implementation of optimal search. Furthermore, the need for a

continuous iterative training and evaluation of the underlying model exacerbate the time complexity incurred during evaluation. As such, in addition to the respective strategy, a computationally efficient exploration should take into account the limitations of the underlying model. To address these limitations, this thesis proposes,

1. The random sparse gaussian approximation method (RSGA). The intuition behind this technique is that, finding the right summarizing points that are part of the training data, is as important as optimising the model parameters. The reason being, the successful evaluation of those points nullify the need to operate on the full data set for pattern discovery. This in return sparsify the resulting model and improve its computational efficiency. The experiments have shown that given that we can find the right summarizing points, the RSGA provides a comparable predictive performance to that of the SVGP, but with a faster computational time. Meaning that optimal kernel evaluation using the RSGA as the underlying model takes the least computational time compared to the other models. Hence, in an effort to reduce the evaluation time while at the same time attempting to secure the simplest kernel combinations, we propose an automatic kernel search algorithm.

2. The stochastic compositional kernel search algorithm. Here, the objective is to utilize the computational efficiency of the RSGA and provide a framework for efficient estimation of suitable kernels. To that end, the method divides and select the available kernels stochastically so that it can map the existing local and global similarities. The experiments have shown that the search algorithm provides optimal kernel mixtures that can appropriately explain the observed patterns. Consequently, it can can be used at the preprocessing stage to determine the best possible kernel combinations. Furthermore, it can also simplify kernel-based learning to fellow researchers who finds kernel selection vague, and to the experienced it can give a preliminary insight into the structure of the data and the possible kernel mixtures that would potentially fit it best.

In both cases stochastic sampling without replacement strategy is followed when building the model for the training point and kernel type selection. The predictive performance of the model is evaluated using the variational gaussian process (VGA) as a benchmark . We run a Monte Carlo type model building with a cross-validation evaluation scheme using the mean square error (MSE) and R^2 score as quality metrics. An ensemble of models were trained and evaluated for different sampling sizes under the same setting. The RSGA approximation gave us a model which is on average 10 times faster than the SVGP. More importantly, a model which makes algorithmic kernel estimation computationally feasible. As a result, we run a stochastic compositional kernel search for suitable kernels that best fit the given data. We applied the root-mean squared error (RMSE) as a criterion to evaluate the optimality of the returned mixtures in explaining the given observation. We tested the algorithm on real and synthetic data. In the experiment we observed, the algorithm offers iteratively possible kernel combinations by following the path with the least RMSE score. The sparsity in model building and the stochastic approach for kernel selection has afforded the algorithm a computational advantage over other exhaustive methods. Hence, for a fast optimal kernel estimation and big data analysis, the RSGA can give an alternative route to model building and inference. Additionally, the search algorithm can be used as an alternative technique for a suitable kernel exploration that best explain the data.

The above approaches only improve the accuracy and computational efficiency of the model using proper kernel selection and sparse approximation. We have already stated how minimum prediction interval and maximized coverage for the forecast points are

relevant features for a given model. For a Gaussian process model, the prediction interval is dependent on the distance between observation and their uncertainty. Meaning that the more data we have, the less ambiguity we will have in model prediction which in return results in a small prediction interval. As a design parameter, the 95% coverage probability is mostly considered for the interval estimation in practice. However, there are no frameworks to check the optimality of the confidence interval and whether or not the forecasted points actually lie within the interval. This is actually true for most models. For example, the Gaussian models are trained using the maximum likelihood as a criterion. Here, the aim is to maximize the likelihood of generating the observed data without considering the quality of the returned prediction interval and/or its coverage probability. Hence, for objective assessment and quantification of the optimality of the confidence interval, predictive models should be trained using the minimum prediction interval width (MPIW) and prediction interval coverage probability (PICP) as performance metrics. With that in mind, this thesis proposes,

1. A distribution based upper lower bound estimation model for deep neural networks using MPIW and PICP as quality metrics. Neural networks are great at making points estimates. However, the lack of uncertainty quantification through a predictive distribution has constrained their application in sensitive areas. Recent advances into deep probabilistic models such as the Bayesian neural network (BNN) have enabled the models to handle uncertainty and provide a distribution. However, like the gaussian process models, the maximum likelihood approach for model fitting presents the same challenge for the optimal quantification of confidence interval. The proposed interval estimation method takes a smaller prediction interval and higher coverage for the predicted points as its performance index. We mathematically demonstrate a distribution based coverage probability and interval width assessment method. The approach directly encompasses the quality metrics in its bound estimation. In addition to that, adaptive hyperparameter is used to weigh the relative importance of prediction interval versus coverage probability. Based on these, a customized loss function is written for model training. A randomized parameter initialization and aggregated ensemble models were considered for the aleatoric and epistemic uncertainty quantification. The performance of the proposed approach was tested on both synthetic and various UCI regression real data sets using the recent quality driven (QD) bound estimation method as a benchmark.

The experiments showed the algorithm can adaptively change the variance of the distribution in order to allow a wider or a smaller sampling areas. Consequently, narrowing and widening the prediction interval commensurate with the needed coverage areas. The algorithm can achieve the desired quality metrics score for data with complex pattern employing a minimum number of layers and neurons as compared to other alternative interval prediction algorithms. It also offers a simple tuning and a stable evaluation for a predictive distribution regardless of the network settings. In the simulation for the same 95% coverage probability, the algorithm achieved a minimum prediction interval. Although, we assumed a logistically approximated Gaussian data distribution during the derivation, the approach has a robust response to data with asymmetric distribution as well. Hence, the algorithm can be used as an alternative learning method to provide a predictive distribution in neural networks.

Motivation

La demande en énergie n'a jamais été un problème pour le secteur de l'énergie par le passé. L'accessibilité et l'abordabilité du pétrole bon marché ont minimisé la volatilité sur le marché de l'énergie. L'indice de croissance démographique bas, l'absence de sensibilisation environnementale et l'industrialisation à petite échelle sont également quelques autres facteurs qui ont naturellement contribué à maintenir l'équilibre entre l'offre et la demande. Par conséquent, la sécurité énergétique n'était pas considérée comme une préoccupation majeure pour la plupart des pays. Cependant, le changement rapide dans l'économie mondiale, l'urbanisation, la démographie et la forte hausse du prix du pétrole ont mis fin à l'ère de l'abondance. Malgré le prix, l'aggravation des conditions climatiques et ses risques environnementaux ont clairement montré que nous devons limiter et minimiser l'utilisation des ressources énergétiques à base de carbone.

Cette tendance vers l'économie verte et la société zéro carbone signifie que la demande énergétique future sera satisfaite par des sources d'énergie respectueuses de l'environnement et pourtant saisonnières. L'intégration de ces sources dans le réseau pour répondre aux demandes quotidiennes et futures nécessite un changement de perspective concernant notre politique énergétique, l'architecture du réseau, la production d'énergie, le stockage, le transport et la consommation d'énergie. Cela crée un hub de systèmes intégrés, dynamiques et complexes, tous travaillant ensemble pour atteindre l'équilibre entre l'offre et la demande. Ces systèmes tels que le réseau énergétique, les réseaux de télécommunication, de transport et autres appareils connectés présentent leur propre complexité et des incertitudes associées. De plus, la nature intermittente des sources d'énergie ajoute intrinsèquement une couche supplémentaire d'incertitude sur la demande énergétique. Même si elle n'est pas encore au niveau attendu, la modernisation du réseau et l'utilisation des ressources énergétiques renouvelables sont en cours dans le monde entier. Cependant, actuellement, la majorité de la demande énergétique est encore couverte par des sources d'énergie carbonées coûteuses.

À une époque où l'énergie est la plus chère, la demande d'énergie de pointe générée en raison de pannes d'usines électriques et de changements saisonniers nuit à la productivité des secteurs manufacturiers et à la rentabilité des fournisseurs d'énergie. Par conséquent, pendant la période de demande de pointe, les entreprises de services publics adoptent le schéma de réponse à la demande qui leur permet de réduire la puissance fournie aux clients plutôt que de l'acheter sur le marché international. Cette stratégie d'atténuation devrait être mise en œuvre de manière à maximiser la rentabilité des clients ainsi que des entreprises de services publics. En conséquence, le succès du paradigme de la réponse à la demande, entre autres, repose sur la précision de la prédiction de la demande et de l'offre d'énergie à l'avance. Actuellement, les modèles paramétriques sont les modèles prédictifs les plus utilisés dans le secteur de l'énergie. Bien qu'ils soient simplistes et efficaces, ils ne peuvent pas gérer les interactions de caractéristiques complexes.

De nos jours, les entreprises énergétiques investissent massivement dans des modèles prédictifs non paramétriques. Les modèles prédictifs qui prennent en compte la volatilité anticipée de la demande et de l'offre pourraient aider les entreprises de services publics à minimiser les risques et à maximiser les profits lors du processus décisionnel. Dans cet

esprit, cette thèse vise à examiner la mise en œuvre de modèles prédictifs probabilistes dans le secteur de l'énergie. Cependant, les incertitudes qui découlent de la nature intermittente de l'approvisionnement en énergie, des interactions entre différents systèmes et du volume considérable de données générées en résultant soulèvent des questions sérieuses concernant l'efficacité de ces modèles. Des questions liées à l'exactitude de la prédiction, à l'efficacité computationnelle, à la largeur de l'intervalle de prédiction, à la probabilité de couverture et, plus important encore, à leur extensibilité aux grandes données. Dans cette optique, cette dissertation vise à répondre à ces questions à travers les objectifs énoncés ci-dessous.

Objectifs

Cette dissertation se concentre principalement sur l'investigation des approches réalisables pour améliorer la précision prédictive et l'efficacité computationnelle des modèles de processus Gaussien et d'apprentissage profond probabiliste. En tant que tel,

- Elle examine la mise en œuvre et l'adéquation des modèles d'apprentissage automatique probabiliste non paramétriques et d'apprentissage profond. Analyse leur efficacité prédictive et leurs performances computationnelles. Et propose une approximation alternative pour garantir leur extensibilité.
- Propose un algorithme de recherche de noyau efficace pour rectifier les contraintes de temps imposées à l'entraînement et à l'évaluation du modèle.
- Propose un nouvel algorithme qui permettra une distribution prédictive pour les modèles d'apprentissage profond.

Principales Contributions

Le processus Gaussien est un modèle prédictif non paramétrique basé sur les noyaux. Son cadre Bayésien lui permet de fournir une distribution prédictive plutôt que des estimations ponctuelles. Cependant, sa performance est contrainte par la taille des données, le type de noyau et le cadre d'inférence. Pour les modèles d'apprentissage automatique, l'inférence et la généralisation sont directement liées à la quantité et à la qualité des données disponibles. Plus de données offrent de meilleures opportunités d'apprentissage. Ce fait est particulièrement vrai pour les modèles non paramétriques, en particulier ceux qui suivent un apprentissage basé sur les noyaux, dont la performance prédictive dépend également du type et du nombre de noyaux sélectionnés. Le processus Gaussien (GP) utilise ces noyaux afin de modeler une distribution a priori sur les données fournies. Par conséquent, sa performance prédictive dépend du type de noyaux sélectionnés pour découvrir les motifs dans les données. La dimension de la matrice de noyaux est un autre aspect qui pose un défi computationnel lors de l'analyse de grandes données. La taille de la matrice de noyaux est équivalente au nombre de points de données considérés pour l'ajustement. Ainsi, l'entraînement des modèles GP sur de grandes données entraîne un coût computationnel et une exigence en mémoire élevés. Cela limite involontairement le modèle sous-jacent. C'est précisément ce fait qui a empêché l'évolutivité du processus Gaussien aux grandes données.

Les difficultés computationnelles étant le principal problème, le succès de l'apprentissage basé sur les noyaux dépend également de la capacité des utilisateurs à sélectionner et à composer des noyaux appropriés. La sélection automatique de noyaux basée sur des algorithmes, bien que possible, est intensivement computationnelle. En plus de cela, les

combinaisons de noyaux augmentent le nombre d'hyperparamètres, ce qui affecte directement l'efficacité computationnelle du modèle. Par conséquent, la mise en œuvre directe de l'apprentissage basé sur les noyaux sur des données complexes a été difficile. Néanmoins, diverses méthodes ont été suggérées pour son estimation optimale. Par exemple, les méthodes de recherche exhaustive, en grille, aléatoire et non paramétrique sont quelques-unes des mentions notables. L'efficacité de ces approches dépend des subtilités des données et des cadres dans lesquels elles opèrent. Par exemple, dans les modèles Gaussiens, la dimension du noyau présente un défi pour une évaluation de noyau appropriée. Dans le cas des modèles basés sur des méthodes variationnelles et MCMC, la complexité temporelle requise pour l'ELBO et la convergence postérieure entrave la mise en œuvre de la recherche optimale. De plus, la nécessité d'un entraînement et d'une évaluation itératifs continus du modèle sous-jacent aggrave la complexité temporelle encourue lors de l'évaluation. En conséquence, en plus de la stratégie respective, une exploration efficace sur le plan computationnel doit prendre en compte les limitations du modèle sous-jacent. Pour remédier à ces limitations, cette thèse propose,

1. Un modèle d'estimation des bornes supérieures et inférieures basé sur la distribution pour les réseaux neuronaux profonds utilisant MPIW et PICP comme métriques de qualité. Les réseaux neuronaux sont excellents pour faire des estimations ponctuelles. Cependant, le manque de quantification de l'incertitude à travers une distribution prédictive a limité leur application dans des domaines sensibles. Les récents progrès dans les modèles probabilistes profonds tels que le réseau neuronal bayésien (BNN) ont permis aux modèles de gérer l'incertitude et de fournir une distribution. Cependant, comme les modèles de processus gaussiens, l'approche du maximum de vraisemblance pour l'ajustement du modèle présente le même défi pour la quantification optimale de l'intervalle de confiance. La méthode d'estimation d'intervalle proposée prend comme indice de performance un intervalle de prédiction plus petit et une couverture plus élevée pour les points prédits. Nous démontrons mathématiquement une méthode d'évaluation de la probabilité de couverture et de la largeur de l'intervalle basée sur la distribution. L'approche intègre directement les métriques de qualité dans son estimation de bornes. De plus, un hyperparamètre adaptatif est utilisé pour pondérer l'importance relative de l'intervalle de prédiction par rapport à la probabilité de couverture. Sur cette base, une fonction de perte personnalisée est écrite pour l'entraînement du modèle. Une initialisation des paramètres aléatoire et des modèles d'ensemble agrégés ont été considérés pour la quantification de l'incertitude aléatoire et épistémique. La performance de l'approche proposée a été testée à la fois sur des ensembles de données synthétiques et sur divers ensembles de données réelles de régression de UCI en utilisant la récente méthode d'estimation des bornes axée sur la qualité (QD) comme point de référence.

Dans les deux cas, une stratégie d'échantillonnage stochastique sans remplacement est suivie lors de la construction du modèle pour la sélection des points d'entraînement et du type de noyau. Les performances prédictives du modèle sont évaluées en utilisant le processus gaussien variationnel (VGA) comme point de référence. Nous avons exécuté un modèle de type Monte Carlo avec un schéma d'évaluation en validation croisée en utilisant l'erreur quadratique moyenne (MSE) et le score R^2 comme métriques de qualité. Un ensemble de modèles a été entraîné et évalué pour différentes tailles d'échantillonnage dans le même cadre. L'approximation RSGA nous a donné un modèle qui est en moyenne 10 fois plus rapide que le SVGP. Plus important encore, un modèle rendant l'estimation algorithmique du noyau réalisable sur le plan computationnel. En conséquence, nous avons exécuté une recherche stochastique de noyaux compositionnels pour trouver des noyaux adaptés qui correspondent le mieux aux données données. Nous avons appliqué l'erreur

quadratique moyenne (RMSE) comme critère pour évaluer l'optimalité des mélanges retournés dans l'explication de l'observation donnée. Nous avons testé l'algorithme sur des données réelles et synthétiques. Dans l'expérience, nous avons observé que l'algorithme offre itérativement des combinaisons de noyaux possibles en suivant le chemin avec le score RMSE le plus faible. La parcimonie dans la construction du modèle et l'approche stochastique pour la sélection du noyau ont offert à l'algorithme un avantage computationnel par rapport aux autres méthodes exhaustives. Par conséquent, pour une estimation rapide et optimale du noyau et une analyse de données volumineuses, le RSGA peut offrir une alternative à la construction de modèles et à l'inférence. De plus, l'algorithme de recherche peut être utilisé comme technique alternative pour une exploration de noyau adaptée qui explique au mieux les données.

Les approches ci-dessus améliorent uniquement la précision et l'efficacité computationnelle du modèle en utilisant une sélection appropriée de noyaux et une approximation parcimonieuse. Nous avons déjà expliqué comment un intervalle de prédiction minimal et une couverture maximisée pour les points de prévision sont des caractéristiques pertinentes pour un modèle donné. Pour un modèle de processus Gaussien, l'intervalle de prédiction dépend de la distance entre l'observation et son incertitude. Cela signifie que plus nous avons de données, moins nous aurons d'ambiguïté dans la prédiction du modèle, ce qui se traduit par un petit intervalle de prédiction. En tant que paramètre de conception, la probabilité de couverture de 95% est principalement considérée pour l'estimation de l'intervalle en pratique. Cependant, il n'y a pas de cadres pour vérifier l'optimalité de l'intervalle de confiance et si les points prévus se trouvent réellement dans l'intervalle ou non. Cela est en fait vrai pour la plupart des modèles. Par exemple, les modèles Gaussiens sont entraînés en utilisant le maximum de vraisemblance comme critère. L'objectif est de maximiser la probabilité de générer les données observées sans tenir compte de la qualité de l'intervalle de prédiction retourné et/ou de sa probabilité de couverture. Par conséquent, pour une évaluation objective et une quantification de l'optimalité de l'intervalle de confiance, les modèles prédictifs doivent être entraînés en utilisant la largeur minimale de l'intervalle de prédiction (MPIW) et la probabilité de couverture de l'intervalle de prédiction (PICP) comme métriques de performance. Dans cette optique, cette thèse propose,

1. Un modèle d'estimation des bornes supérieures et inférieures basé sur la distribution pour les réseaux neuronaux profonds utilisant MPIW et PICP comme métriques de qualité. Les réseaux neuronaux sont excellents pour faire des estimations ponctuelles. Cependant, le manque de quantification de l'incertitude à travers une distribution prédictive a limité leur application dans des domaines sensibles. Les avancées récentes dans les modèles probabilistes profonds tels que le réseau neuronal bayésien (BNN) ont permis aux modèles de gérer l'incertitude et de fournir une distribution. Cependant, comme pour les modèles de processus gaussiens, l'approche du maximum de vraisemblance pour l'ajustement du modèle présente le même défi pour la quantification optimale de l'intervalle de confiance. La méthode d'estimation d'intervalle proposée prend comme indice de performance un intervalle de prédiction plus petit et une couverture plus élevée pour les points prédits. Nous démontrons mathématiquement une méthode d'évaluation de la probabilité de couverture et de la largeur de l'intervalle basée sur la distribution. L'approche intègre directement les métriques de qualité dans son estimation de bornes. De plus, un hyperparamètre adaptatif est utilisé pour pondérer l'importance relative de l'intervalle de prédiction par rapport à la probabilité de couverture. Sur cette base, une fonction de perte personnalisée est écrite pour l'entraînement du modèle. Une initialisation des paramètres aléatoire et des modèles d'ensemble agrégés ont été considérés pour la quantification de l'incertitude aléatoire et épistémique. La performance de

l'approche proposée a été testée à la fois sur des ensembles de données synthétiques et sur divers ensembles de données réelles de régression de UCI en utilisant la récente méthode d'estimation des bornes axée sur la qualité (QD) comme point de référence.

Les expériences ont montré que l'algorithme peut adapter de manière adaptative la variance de la distribution afin de permettre des zones d'échantillonnage plus larges ou plus petites. Par conséquent, il réduit ou élargit l'intervalle de prédiction en fonction des zones de couverture nécessaires. L'algorithme peut atteindre le score des métriques de qualité souhaité pour des données présentant un motif complexe en utilisant un nombre minimal de couches et de neurones par rapport aux autres algorithmes alternatifs de prédiction d'intervalle. Il offre également un réglage simple et une évaluation stable pour une distribution prédictive quel que soit le paramétrage du réseau. Dans la simulation pour la même probabilité de couverture de 95%, l'algorithme a atteint un intervalle de prédiction minimal. Bien que nous ayons supposé une distribution de données gaussienne approximativement logistique lors de la dérivation, l'approche présente également une réponse robuste aux données présentant une distribution asymétrique. Par conséquent, l'algorithme peut être utilisé comme une méthode d'apprentissage alternative pour fournir une distribution prédictive dans les réseaux neuronaux.

Outline

In line with our objectives, this dissertation evaluates the performance, limitation and relevance of non-parametric models to the energy sector. With that in mind, the thesis is organized into 5 chapters. The contribution of each chapter is outlined below,

Chapter 1 presents an introductory reviews on the changing landscapes in grid evolution, challenges to a large scale utilization of renewable energy sources and future opportunities in grid modernization. Furthermore, we will review the contributions of probabilistic models to the energy sector. More importantly, how their deployment could help in managing the uncertainties that could arise during the introduction of variable sources to the grid.

Chapter 2 takes a deep dive into probabilistic models, Gaussian process and its approximates, such as, the deterministic training conditional (DTC), the fully independent training conditional (FITC) and the sparse variational gaussian process (SVGP). We will also review in detail, the kernel basis functions which characterizes the dynamics of the Gaussian models and see how their selection affects the prior and posterior distribution of the resulting model. In addition to that, we will examine some of the approximation methods that are followed in order to improve the computational efficiency of the model and its scalability to large data. Specifically, we will review model approximation through pseudo points and variational inference.

Chapter 3 evaluate the implementation of SVGP on electricity demand profile data and review its limitation. To improve the performance of the model, the chapter introduces a stochastic kernel estimation algorithm. We will also propose the sparse Gaussian approximation based on random column sampling. This model will be used as the underlying model for running the kernel search algorithm. The performance of the proposed techniques will be evaluated in detail.

Chapter 4 will review uncertainty quantification in the context of deep learning framework. Here, we present a method for better uncertainty quantification. We will also see the importance of defining coverage probability and prediction interval as performance metrics. Based on that, the chapter introduces a distribution based lower upper bound estimation algorithm. The performance of the algorithm will be evaluated against the latest bound estimation algorithm using real and synthetic data. And finally,

Chapter 5 will revise and conclude our contribution in relation to forecasting in the energy sector, machine learning and future research direction.

Chapter 1

Predictive Models in Energy Sector

1.1 Introduction

We have heard the benefits of electricity in one way or another from different sources. A mere visualization of modern life without electricity can give a better contextualization for its importance. Human civilization is built on top of our ability to cooperate and communicate. The computer-networks that allowed us to process and exchange complex ideas, the machinery that made us productive in farming, mining, industries, the equipment in science and education that gave us insight and improved our understanding, the home appliances (i.e heating, refrigeration, air conditioners) that increased the comfort and convenience of life, are all powered by electricity. That is why the consumption of electricity is considered as a good indicator for the quality of life [1, 2]. Furthermore, improving the accessibility and affordability of electricity could enhance the development of a country by driving the migration of companies and industries towards it. As such, ensuring the reliability, affordability and sustainability of energy production has been the goal of any aspiring nation. These commitments has never been more apparent than in Europe where the new ITER fusion reactor is being built.

The individualistic and nationwide competition for the acquisition of energy has driven its demand to grow exponentially. The growth in population, urbanization, consumerism and decarbonization through electrically powered transportation systems are the key drivers for the rise in demand. Consequently, the finances involved in building, transportation and maintenance of the electric grid have made the price of electricity expensive. The pricing is also dependent on the type of resources used for power generation and more importantly who controls it. The Russian-Ukrainian confrontation and the surge in the price of electricity in Europe can be taken as a good case in point on how political dynamics influence the price [2, 3]. As a result, for most countries energy self-sufficiency has been a top priority. These in turn has resulted in a wide penetration of renewable energy resources (RER).

Technological and manufacturing improvement in the areas of wind and solar energy has reduced their cost and increased their deployment. The international energy agency (IEA) projection in 2021 indicated that solar energy utilization is expected to reach closer to 3 terra watt (TW) and wind energy will surpass 1.5 TW by 2030 as shown in Figure 1.1. This has exceeded the 2006 world energy outlook prediction for RER utilization by 2030. Solar is expected to show 30 fold increment whereas wind energy share will quadruple compared to the initial extrapolation in 2006. Further improvement in manufacturing, wind turbine design, photovoltaic and thermal energy conversion technology, finance and wider public support could make their energy share even more by 2030 [2, 3, 4, 5].

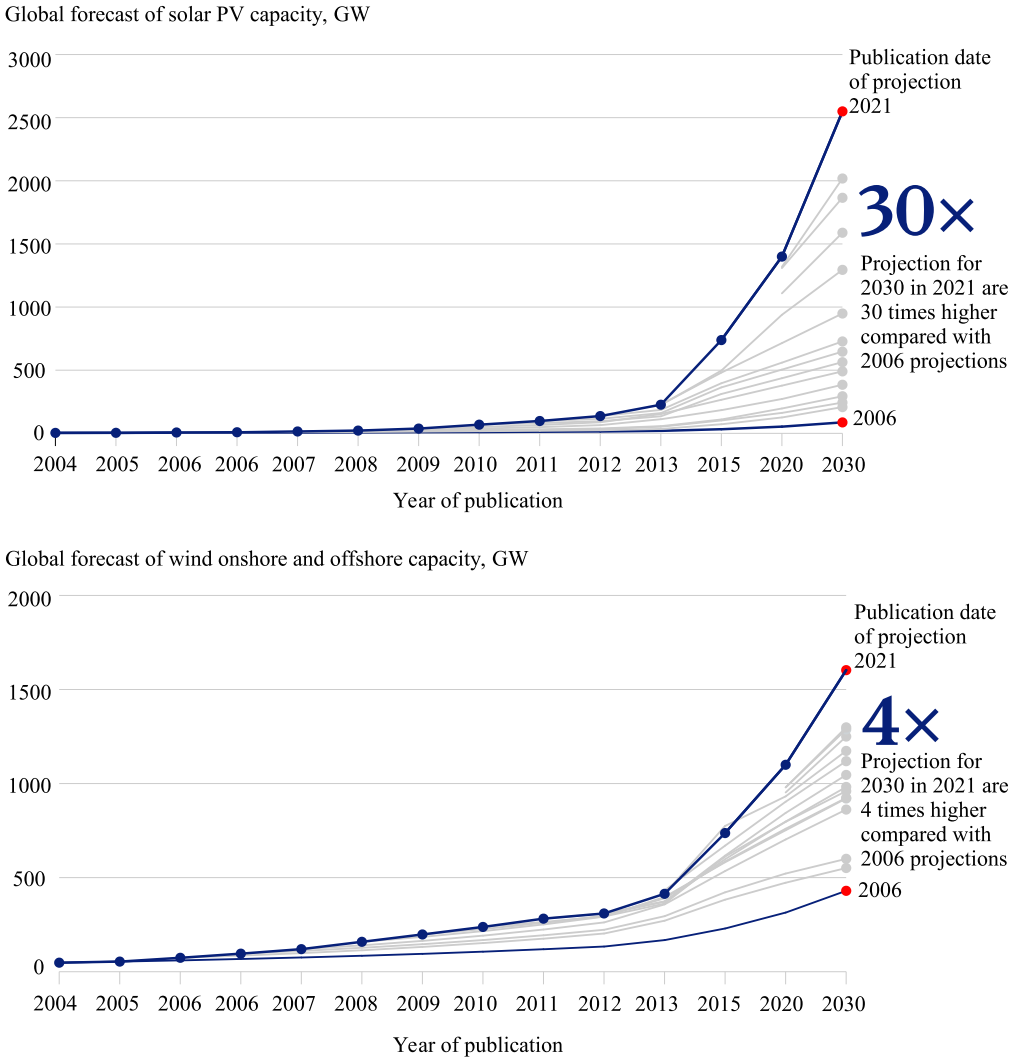


Figure 1.1: Renewable energy resource utilization projection for the year 2030 according to IEA. *Source: IEA renewable data and Mckinsey edition on Renewable-energy development in a net-zero world*

1.2 Challenges in renewable energy deployment

For the past few years, the demand for renewable energy resources has dramatically exceeded all expectations. Further improvement in manufacturing, wind turbine design, photovoltaic and thermal energy conversion technology, finance and wider public support are set to make their energy share even more [2, 3, 5]. However, a successful transition to clean energy is not as easy as one could imagine considering their obvious advantages. Hence, a complete shift requires addressing the social and technological challenges we are facing today. Furthermore, it needs the full participation of policy makers, companies and citizens. Consequently, to speed up this transition, few key challenges must be addressed.

1.2.1 Society

As much as people are eager to enjoy the benefits of electricity and despite the impacts clean energy has on combating climate degradation, some people are not open to the idea of large scale RER deployment in their backyard. Some of these concerns may be warranted. For instance, the construction of a hydro-power reservoir can have a negative

impact on the environment or the livelihood of those around it. The site could cover agricultural and historical areas, homes, forests or could obstruct the migration of fishes [6] which might be a source of income or sustenance to those living downstream. In the case of solar and wind farms, a change of landscape scenery, noises, moving shadows and flickering lights are some of the reported concerns [2]. Despite the inconvenience, people are more ready to make a concessions or compromise when they are properly communicated on the pros or compensated for their trouble. Consequently, one approach that is suggested as a mitigation strategy is to incentivize the community. Involving individuals on large scale projects as direct participants in owning share of the project, could empower and make them more amicable to the idea of deployment [2, 3]. Hence, RER deployment should involve a careful design considerations in rectifying the adverse effect it has on the environment and societal well-being. Most importantly, it should provide a positive contribution to the social and economic development [10].

1.2.2 Governmental policies

Energy from carbon based minerals (i.e natural gas, coal, oil) comes from a finite resources which aren't available in ample amount at every corner of the world. This scarcity is also evident in some of the renewable energy resources. As an example, energy harvest from a hydro-power is dependent on the location and accessibility of enough water. Not all places are fit for Hydro-power generation. In contrary, solar and wind power can be harvested sufficiently everywhere. Consequently, they account for most of the government energy policy toward renewable energy deployment [2, 3, 10]. RER utilization is growing, perhaps not as much as expected or not at the same scale everywhere. Currently, Asia takes the lion share in RER expansion. However, a faster transition to clean energy requires the participation of the government in ratifying flexible policies capable of tackling the private and public regulation barriers that hinders RER deployment.

Policies that motivate the use of carbon-based energy resources such as government oil subsidization or preferential tax treatments that lower taxation for oil importers must be discouraged [10]. On contrary, policies that make investment into large scale renewable installation attractive should be appreciated. Reforms that create a conducive environment for investors to involve in renewable projects, such as, tax incentives in the form of deductions, credits and exceptions should be promoted. The government should promote grants and subsidization for individuals willing to invest or utilize renewable energies. Energy policies such as the feed-in tariff (FITs) payment scheme that guarantee a return of investment to potential investors should be encouraged [3, 10].

Investment goes hand in hand with public awareness. Investors are more inclined for a risk when there is a demand for it. As such, the government should promote a wide public awareness on the explicit reasons for the shift toward a clean energy and its potential benefits to the general public [10]. On the other hand, the government should make sure that energy providers strictly and consistently follow the renewable energy portfolio standards (RPS). In RPS market based policy, energy providers are required to diversify and supply a portion of their energy production from renewable energy resources [3]. Adherence to these policies will promote a gradual shift toward a sustainable clean energy utilization. Reform policies only reflect the government willingness. However, for a true transformation these policies must be supported by grid modernization. One aspect of renewable investment that makes it off-putting to potential investors is that it is more capital intensive compared to the carbon based energy investments [7]. Hence, grants and support in grid modernization will make investment in renewable energy attractive and integration easier.

1.2.3 Technology

Research and development in material science and manufacturing has made RER deployment cost-effective. This has enhanced their wider penetration and increased their global average energy share [7]. However, import-export taxation on renewable energy equipment's could add unnecessary additional finance, reduce affordability and hamper wide scale utilization. Hence, governments should focus on producing solar panels, wind turbines and all the necessary accessories locally for renewable energy extraction. Financial grants and support for homegrown R&D and industrial expansion can potentially relieve the pressure on countries dependence on imported accessories and increase renewable energy affordability. Solar and wind energy extraction is a matured science. However, there is still a long way to go in terms of revolutionizing the design, efficiency of energy extraction and storage. Hence, governments continual support to R&D must be a priority [10].

Technological advancement should also focus on knowledge transfer. Renewable infrastructure production without installation doesn't bring the intended result. Most people even if they are inclined to the idea of renewable energy utilization, they face technical difficulties during installation. For instance, technical difficulty's such as grid incompatibility, location inaccessibility for installation, operational acumen, energy market penetration and competitiveness. Hence, the government should also focus on capacity building and skilled workforce in the areas of renewable energy [2].

1.3 Traditional grid

Renewable energy integration is one aspect that should be considered for a full scale deployment. Energy production is one thing, but to be useful it needs to transported. For a community participation as both energy producers and consumers, the existing grid requires modernization. This has been a common concern for a renewable energy investment. Additionally, their vulnerabilities to a periodical reduced power generation puts their dependability to question. Sometimes, this reduction compels the utilities to use a more expensive sources of energy. Thereby increasing the price of electricity. Ironically, the pricing mostly affects the commercial and residential consumers. Industrial consumers enjoy the privilege of receiving electricity at a higher voltage and lower current which minimizes the transmission loss. Hence, they receive electricity at a lower cost per kilowatt-hour compared to other consumers [8].

The difference in the payment is attributed to the existing grid architecture. The traditional grid has been the backbone of technological advancement for hundred's of years and it will be so for sometimes in the future. By design, it consist a centralized large scale generators stationed far from the consumption site considering their environmental impact, resources availability, cooling so an so forth, as shown in the left part of Figure 1.2. Such arrangement helps the utilities to generate power for large consumer bases at a time. This will nullify the inconsistency in individual energy consumption and flattens the average energy requirement. Inherently, this improves the predictability of the demand. As such, load anticipation and optimal generator dispatch will be easier. However, their distance from their consumer base, significantly increase the amount of energy wasted on the transmission line. Hence, more energy is produced than needed to compensate for the transmission loss. Unfortunately, the consumers are expected to pay both for their energy demand and the averaged loss. Additionally, the existing grid architecture is not scalable to the ever increasing energy demand.

As the demand grows, the traditional mantra dictates large and large power plants must be constructed [8]. And, this has been the trend for a hundred of years. The build and forget approach for energy generation and transmission is no longer sustainable.

Especially in this era, where we have witnessed a tremendous need for energy due to the rapid population growth, urbanization and a step by step inclination towards a carbon free society such brute force approach is not economical. As such, building a bigger power plants for every foreseeable demand is not financially viable (sound). Furthermore, any natural or man-made incidents could put the whole grid at risk of a shutdown [8].

In the event of a single point of failure, a centralized network exposes itself to a cascaded failure that sometimes results in a complete system shutdown. Thereby impacting its reliability. More importantly, the existing grid has a limitation for a full renewable energy integration. Grid expansion, communication devices, smart meters, and other extra accessories are required for a distributed renewable energy integration. As a result, these limitations demand a complete paradigm shift in the gird architecture that guarantee the sustainability, dependability and efficiency of how energy is produced and consumed. And, possibly result in a profit gain for all parties involved in electricity production and consumption [8].

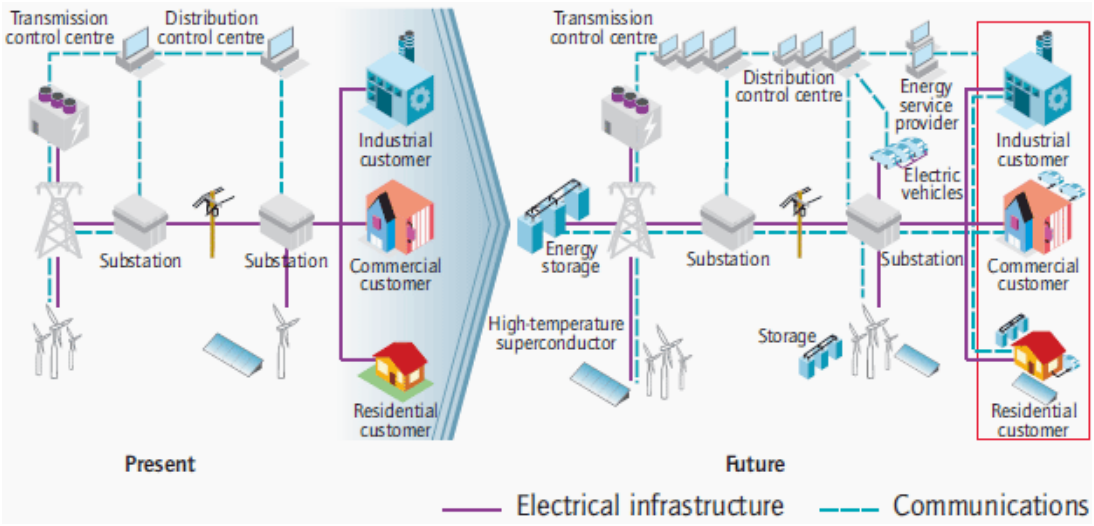


Figure 1.2: Grid genesis from traditional to smart infrastructure. *source: smart grid concept and characteristics, a technical article by Edvard Csanyi*

1.4 Smart grid

The notion of smart grid has been around for quite sometime. However, in many parts of the world, a complete topological shift from the centralized grid to a smart grid networks is just taking a root. Smart grid topology is considered as the next evolutionary step to the traditional electric grid. Both implicit and explicit reasons can be given as to why this shift is necessary at this age. But a better contextualization for its necessity can be stated by examining some of the shortcoming of the existing network. Here, we would like to make it clear that when we say by shortcomings we are not claiming these issues we are about to discuss render the traditional grid obsolete. But rather, they make the grid inefficient in more ways than one. As such, they should be considered as a mere limitations.

The promise of smart grid is great and especially the economic incentives for the participants are what makes it truly revolutionary. Smart grid relies on a smaller decentralized generation units based on renewable energy resources and located closer to the consumer sides as shown in the right side of Figure 1.2. The concept of decentralization creates a more resilient and self-healing grid architecture. Smaller units offers a production affordability potentially turning consumers into intermittent energy suppliers. Meaning that the

uni-directional communication/energy flow from producers to consumers is now changed into bi-directional. The production units are located closer to the consumption sites thereby minimizing transportation loss and improving energy utilization. Furthermore, renewable energy resources utilization have increased the accessibility and affordability of cheap electricity to many. However, its implementation requires complex infrastructure compared to the traditional grid. At every stage it requires the integration of different automation, communication and computer systems components to deliver power from the generation cite to consumption which resulted in the generation of big data [11].

As it is composed of a smaller distributed units, the convenience of bulk power estimation is no longer valid. Consequently, there is high variability both in energy production and consumption [8]. As such, a successful demand response requires better predictive models at the either ends of the power flow. These predictive models should address the variability of supply and demand, handle the big data generated and a provide a marginal profit gain for all parties involved. This is where we believe probabilistic models will play a vital role for a successful implementation of demand response optimization.

1.4.1 Smart grid challenges

Grid transformation is something that should be a priority for a successful distributed energy assimilation. This evolution requires the integration of communication technologies, smart meters and control infrastructures, protocols and architectures to handle the large amount of data that is generated into the existing grid. However, a number of challenges needs to be addressed before that happen. The size of data implies the grid needs to be scalable to the ever increasing customers number. As more and more customer who potentially could become producers themselves and who employ different metering and generation sources, a lot of the data is generated. This data could create an issue with latency, bandwidth and reliability. Furthermore, with connectivity and automation comes a security breach [8].

Some of these security breaches include data and password theft, denial of service, grid violations so and so forth. These attacks can range from energy theft to grid instability resulting in localized or a complete system shutdown and generators damages [11, 12]. The impact of these violations has already been witnessed in the Dec 2015 power grid attack in Ukraine, in the Dec 2014 steel mill attack in Germany, in the Mar 2021 electric company attack in Australia and int the Jan 2020 electric corporation attack in Israel. Even worse, the internet of things that is yet to come will open the gateway so that the type and manner of these attacks will be more frequent. As such, safeguarding the grid and the users information from possible cyber crimes must also be a priority. Hence, the protocols, architectures, communication and grid specific devices should be designed to address the volume of data, possible security vulnerabilities and the associated mitigation strategy in mind [3, 11, 12].

1.5 Role of predictive models

Today's energy market is demand-driven where customers consumption dictate the amount of raw materials required for power generation. Hence, energy is generated in real-time to meet the existing need. Inherently, this demand is not constant and is continuously changing through out the day [8]. It goes in a periodical peak and off-peak cycle depending on the time of the day and day of a week. As such, energy providers prefer a more controllable resource for demand supply equilibrium. This fact has made carbon-based raw materials prime candidates. Among the renewables only few present themselves controllable like hydro-power. Solar and wind are not controllable. They are a "what you

see is what you get" type of energy resources regardless of the existing demand. Hence, their energy output is entirely dependent on the weather [13].

Currently, the weather affects only our energy consumption. However, when smart grid is up and running, the weather determines not only how we consume energy but also how much we can generate from it. Consequently, the energy exchange will be determined by the weather. Hence, the demand, low or high, doesn't change the amount of available energy or impact how much energy can/should be harvested. This makes balancing energy consumption with production difficult in real-time which puts the stability of the electric grid at risk [15]. Thus, demand-supply equilibrium will be dependent on our ability to understand and predict these intermittent energy resources. In these supply-driven energy exchange, predictive models will help us in planning and managing the move toward clean energy by ensuring renewable resources takes the lion share in the coming paradigm shift [3, 13, 14].

The presence of uncertainty in the supply requires greater control over energy generation and consumption. This can be achieved through active engagement with the customers so that energy is efficiently generated and consumed. The participation can range from adjusting customers demand commensurate with production to allowing the energies providers a partial or full control over their equipment's [8]. The smart grid architecture makes it possible for customers to be both consumers and producers. Hence, at the time of peak-demand, the customers can use their own supply to nullify their energy deficit or become providers themselves. Consequently, predictive models are used to anticipate the variation at both end of the power-flow so that the necessary adjustment can be made before hand. As such, it is fair to say that the success of the balance and grid stability is predicated upon the accuracy of prediction.

The benefits of predictive models goes beyond securing grid stability. They have been used in different sectors for optimising operations that result in efficient utilization of energy and financial gains. For instance, the profitability of an energy company is dependent of the level of customers satisfaction. A consistent power delivery in the event of storms, thunders and other natural phenomena increases the dependability of the company [9]. Thus improving customers trust that results in a higher subscription rate and profit gain. Failing to deliver power during natural disasters creates a dissatisfaction among its customers. Hence, by using predictive models, energy companies can determine the probability of those events happening and plan a predictive maintenance or arrange human resources for immediate intervention in the likely event of a blackouts. Arbitrary maintenance schedules can also be avoided by using predictive models.

Forecast models can show the time of a day and day of a week where there is a least demand so that scheduling can be done without creating inconvenience to the customers. They can also be used in short, medium and long term planning [14]. For instance, in all phases of planning the company can determine the future forecasted demand beforehand and economically acquire the necessary resources that will enable it to generate power consistently without interruption. Demand forecasting can also be clustered and used to identify dense customers areas or new potential areas for future expansion so that the company can use it to setup a facility nearby. This enables it to minimize the infrastructure cost, transmission loss and maximise revenue returns. In a company's day to day operations, forecast models can help avoid over or under power generation by optimising generator dispatches so that equivalent power is generated to stabilise the grid and meet the demand [14].

In building temperature control, predictive models have been used to optimize energy utilization by actively regulating the building thermodynamics without impacting the comfort level of the occupants. It can also show building managers possible areas where energy minimization is needed thereby lowering operational cost.

In industries, predictive models can be used to determine possible areas for trimming energy consumption and how best to schedule machinery's in a way that maximise productivity and minimise energy expenditure. In the event the company is equipped with a local energy production or storage facility, the models could help in determining when to connect to or disconnect from the grid and when to charge or discharge the local storage in a way that minimise energy expenditure and maximise profit. Regardless of the sectors in which they are used, the main objective of predictive models remains the same, ensuring efficient utilization of energy that results in a profit gain for all parties involved in production, transportation and consumption of energy. Their role will be even more paramount and significant in the next evolution of energy exchange and their accuracy will guarantee the successful implementation of smart grid and the wide integration of renewable energy resources.

1.5.1 Predictive models in demand forecasting

Energy companies have used short, medium and long-term demand forecasting to plan, produce and optimise the efficient utilization of energy [14]. Various statistical, AI and hybrid models that differ in nature, methodology, performance and output distribution have been proposed and utilized to that end. Even now, they are investing a lot on new predictive algorithms. A number of researchers have tried to assess the quality of these models objectively. Yet there is no uniform consensus as to which model is best. And we won't be making one. The reason being the predictive performance of a given model is dependent on many factors. For instance, the availability and quality of data, the modeling approach, forecast horizon, hidden associations between predictors, sudden environmental changes, socioeconomic and political dynamics are some these factors that can impacts predictive accuracy. Even then, there are some performance metrics that are inherently the defining characteristics of the model. For example, machine learning models tend to require more computational effort and consume a lot of memory compared to other statistical models. If these were the criteria in which the models were compared with then it would be fair to say that statistical models are better.

A subjective bias introduced by the researcher when selecting the data generating model and tuning its parameters, is also another factor that will skew the assessment. Consequently, different levels of compromises has to be made to select a model that fit the given problem. This makes the very definition of objective assessment vague and the accompanied result subjective. As such, although performance analysis is possible, model selection, the acceptable degree of compromise with regard to its complexity, reliability, accuracy and computational resource requirement should be dictated according to the need and the problem at hand.

Currently Energies companies use a combination statistical (i.e ARIMA, SARIMA, ES, MA, KALMAN,GPR, LR,...etc) and deep & machine learning (i.e MLP, CNN, RNN,GPR, LR,...etc) forecast model suited to their need and forecast horizon. However, a converging research interest is making the distinction between statistical and machine learning models difficult. Hence, it is becoming more common to see a hybrid model that contains the best of both models. In Figure 1.3, we have tried to outline some of the models employed by these companies for demand forecasting. However, as the choice tends to be subjective and tailored to the needs of the company, the list will be very large.

1.5.2 Why probabilistic models?

Humanity has always looked into the future for strategic advantage, for assurance and survival. There are those who argued our desire to look into the future is rooted upon

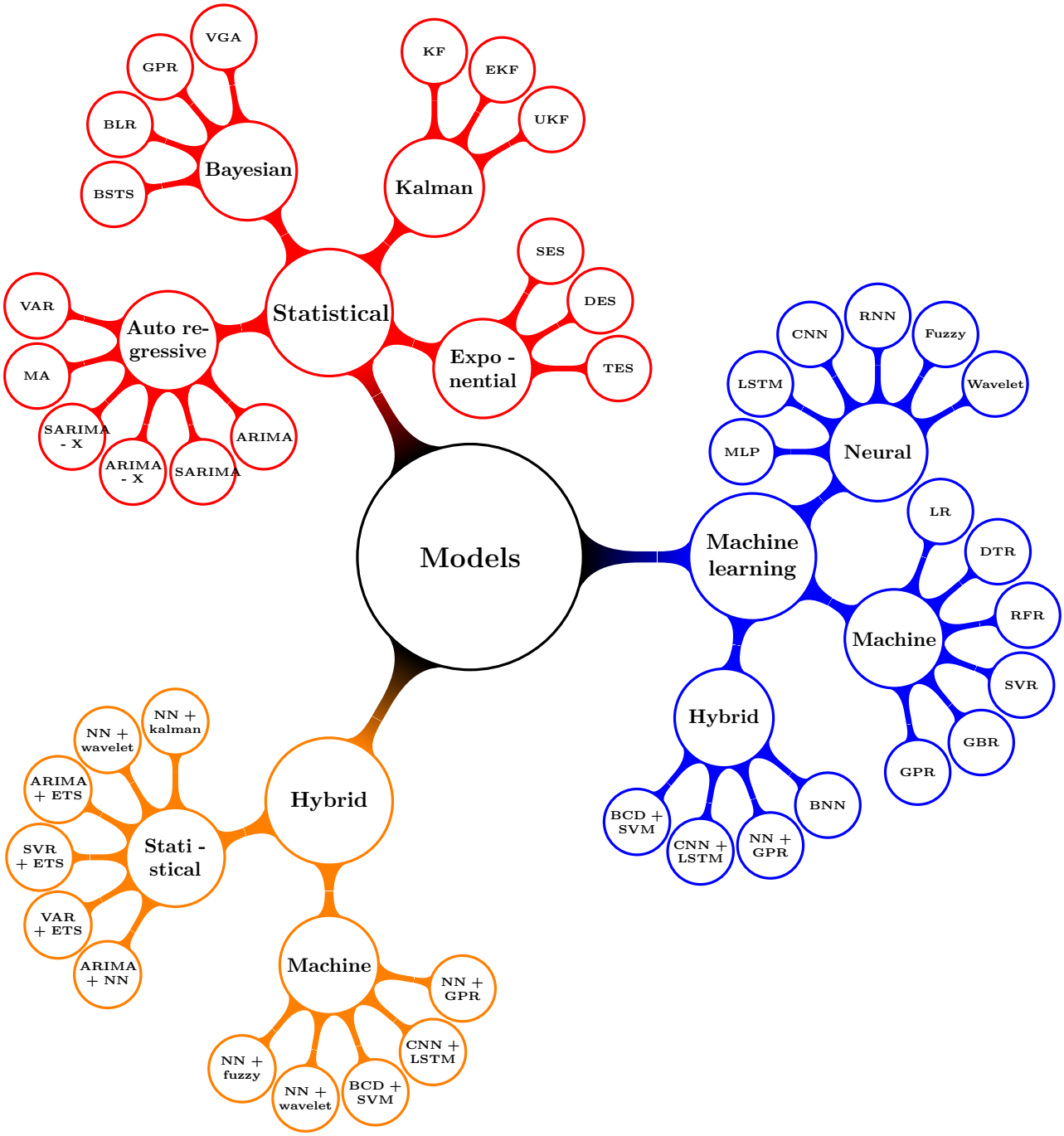


Figure 1.3: Sample electricity demand forecasting models

our perpetual fear of death, destitution and unforeseen calamity [16]. The desire to know the shape of things to come doesn't necessarily mean we will like whatever it has in store for us. In time of uncertainty, good predictions are considered as good omen and filled us with the prospect of hope. And bad predictions left us in despair. Regardless, humanity has shown the willingness to see the future and the responsibility to bear its outcome. For some, predicting the future transcends gaining strategic advantage, assurance or securing once survival.

Martin on his book on "seeing the future" [16] stated that without the desire to know the future "Goals cannot be established, nor efforts towards realizing them launched; nor the consequences of reaching, or not reaching, those goals be considered. Neither can threats and dangers be identified and either met head on or avoided. All this is as true today as it was when we first became human". He argued that without foresight, anticipation or the will to know the future, human life is meaningless [16]. Considering

where we are now compared to where we once were, it is a valid argument. Humanity has done a lot for the sake of self-preservation and fear of the future. The tools we built to protect, the techniques of farming to feed or the rule of low and system of governance that created a society with a common goal, were to ensure our individual and collective survival [17]. But our desire to know the future has exceeded the instinct to survive. Now we look to the future for becoming. As long as we hold on to the idea of becoming better than we were yesterday, humanity will never stop looking into the future. At the age when complex ideas seemed incomprehensible and thought they should be left to the gods, we prayed and believed in beings greater than ours. The desperate ones believed in the goddess of fortune (Fortuna) in the hope of receiving a blessing, those who longed for understanding prayed to Minerva, the goddess of wisdom [18]. Now we are at the age of understanding the nature of things which were once thought to be within the domains of the gods. Through probability and statistical reasoning, we have tried to find meaning in the randomness of chances and entered the realm of Fortuna. Through engineering, we have acquired the eyes and wings of Horus, created the thunder of Zeus, what more is there to the sun god than welding nuclear energy?.

Every scientific research and technological advancement is bringing us an inch closer to becoming better than we were yesterday. There is a lot to be done and humanity is billions miles away from fulfilling its objectives and attaining true understanding. The aspiration to forecast is no different. It is just one goal on a long list of ambitions. Of course, at this age we won't base future decision by looking at the guts of animals like the ancient Romans used to do. Instead we learned to look at data, however, the desire to know the future remained similar regardless. We recognized that tomorrow has a history. If it is recorded, it can be analysed and its future outcome will be deterministic to a degree. To that end, both statistical and machine learning based deterministic models were used to create a possible data generating model that could simulate the observed data. Our experience with deterministic models have showed us, the past is not enough. Things that haven't happened yet could change the outcome. For instance, the rapid technological advancement in the areas of renewable technology has forced the prediction for a renewable energy penetration by the year 2030 to be corrected repeatedly year after year as shown in Figure 1.1.

One factor that could be considered a limitation in deterministic model prediction is their absolute belief in the accuracy of the input data or model parameters. Deterministic models don't leave room for doubt. They don't question the certainty of the data they process or the parameters of the model. For them the past explain the future. There is no questionable difference between prior data and what comes after it [19]. For these models, given the data and model parameters, the answer they provide is 100% correct. They are more certain about the outcome even when they are wrong. They ignore the fundamental fact that all data contains uncertainty whether it is from approximation, systematic or randomness. Such models present potential risks during important decision making.

These uncertainties can emanate from the very system we are trying to control or they can be external to the system. Either ways, in the presence of these uncertainties we can not completely be sure if our estimation is correct. We can take our recent covid pandemic as a good example as to how environmental factors affect the accuracy of a predictive model. The likelihood of this event was closer to none. And yet, despite their parametric nature, most of the predictive models utilized by the energy companies in Europe, couldn't anticipate the sudden drop in electricity demand. Miscalculation will be even more damaging to a cascaded system where the decision of the later sub-system is dependent on the output of the former. Such arrangement is prone to error compounding. If the system manipulate deterministic predictive models for decision making, a propagating error will have a profound effect on its accuracy.

In contrary, a probabilistic model considers all variables and system interactions as random variables that are bound to vary. Therefore, it offers a full spectrum for a possible future outcomes taking into account the inputs randomness. As such, it provides a framework for uncertainty propagation so that all subsequent subsystems work on a probable outcomes rather than a singular value which may or might not be the true value. This enables the system to make an informed decision considering the inherent uncertainty thereby making it more robust and reliable. Consequently, the tendency to include uncertainties leaves probabilistic models at a far better position in forecasting the future than deterministic models. There are a lot of probabilistic models and various methods of accounting for uncertainties. As a result, the question of which one is better is still debatable. However, at the bare minimum a good predictive model should exhibit a minimized prediction interval (PI) and offer maximum coverage. In the coming chapter we will have a lot to say about probabilistic models, prediction interval and coverage probability.

1.6 Conclusion

In this chapter, we discussed how electricity is waived into every aspect of our life. And without it everything we take for granted will be no more. If it hadn't been for the dependence of modern life on electricity, smart grid or grid modernization wouldn't have been an issue. Restructuring the existing grid for monitoring the interconnected components, optimizing the delivery and consumption of electricity and managing distributed power plants is not a small feet. Its realization will ensure the reliability and efficient utilization of energy. However, this grid revolution, although beneficial, it also exposes the grid to a new host of man-made and natural threats. Especially, the integration of weather dependent renewable energy sources by itself put another layer of challenge. These variable sources despite their limitation, they offer the best chance in increasing the affordability of electricity and ensuring the security and energy independence of a country. Consequently, in addition to a wider public awareness and governmental support, smart grid requires innovative technologies in the areas of improving the reliability of delivery, energy generation and storage, efficiency of consumption and a foolproof network architecture to rectify cyber vulnerabilities and the mitigation of the variable energy generation. As part of the mitigation process, predictive models play a vital role in anticipating the changes in the demand for energy and the production capacity so that appropriate demand-supply optimization can be carried out.

In this thesis we considered a probabilistic based machine learning models for demand forecasting mainly for two reasons. First, probabilistic models enhance decision making at the either end of the power flow under uncertainty. Hence, by improving accuracy, it results in a financial gain for all parties involved in generation, transportation and consumption of electricity. Second, grid modernization has generated huge volumes of data and machine learning models are better at analyzing patterns and making sense of big data. To that end, we selected the Gaussian process and its derivatives, like the DTC, FITC and the SVGP as probabilistic model from machine learning, and the neural architectures like LSTM and MLP as a representatives to as a deep learning frameworks. In the coming chapters we will discuss the implementation of these models for demand forecasting in the energy sector in detail.

Chapter 2

Literature Review

2.1 Introduction

In the last decades, energy companies have been investing a lot in predictive algorithms that have different performances according to their complexity, prediction accuracy and computational resource requirement. The performance of these algorithms are dependent on many factors. Apart from their inherent parametric form, the quality of data, prediction horizon, hidden associations between predictors, a sudden socio-economic and environmental changes are some other factors that can impact the quality of forecast. Even the act of stretching the forecast horizon induces the propagation of more uncertainties which restrict the validity of the forecast. Regardless of that, a predictive model should exhibit few desirable features such as a minimized prediction interval (PI) to account for uncertainties and maximized accuracy for mean trajectory. And, a model that combines accuracy with a principled approach to uncertainty quantification through PI enhance the validity of its forecast [20, 21].

In this chapter we will review parametric and non-parametric probabilistic predictive models. More importantly, we will focus on the Gaussian process regression and some of the approximation methods utilized for its scalability to big data analysis and prediction. In chapter 4, we will review probabilistic forecasting in deep learning context.

2.2 Probabilistic predictive models

There are different ways of categorizing a predictive model. They can be classified based on their task, learning approach, method of prediction, nature of their output, so and so forth. For instance, based on their inherent mathematical form, predictive algorithms can be categorized as parametric [22, 23, 24, 25], semi-parametric [26, 27, 28, 29] and non-parametric [21, 20, 30, 31, 32, 33, 34, 35]. Parametric models have a fixed parameter space which make them simple, elegant and interpretable. They allow a reasonable assumption to be made about the structure of the learning model and number of parameters needed. Once fixed, they won't change their mind easily about how many parameters are needed to fit the given observation, despite the amount and complexity of data available [36]. Hence, they generalize the data with a fixed set of parameters regardless of its nature. However, this rigid parameter assignment is restrictive which ultimately constrain what can be learned from the data. Consequently, the understanding of the model doesn't grow as new information is made available. As such, they are highly influenced by unmodeled dynamics [29]. However, they can also be trained to capture relevant information by introducing non-linearity through appropriate transformations. This can be achieved by

modifying the initial assumption on the nature of interaction between the independent variables. The model can be made to learn complex patterns through parameter relaxation, augmentation, adding extra polynomial terms, auxiliary features, or non-linear interaction between independent variables [37]. It should be noted that the extra non-linearity is added as part of the variables modification and for mathematical convenience the model will still remain linear in its parametric representation. Such modification has already been implemented to asses the effect of environmental dynamics on the predictive performance of a model. For instance, the generalized additive model with augmented adaptive parameter estimation using kalman filter was suggested as a mitigation strategy to improve responsiveness and forecast accuracy during the covid pandemic [38].

On the other hand, non-parametric models have unconstrained parameter space. These are the models where we don't make any assumption about the number of parameters required to fit the given data. Instead, the objective is generating a function that is consistent with the observation. This allowed them to change their function space commensurately with the data which in return improved their generalization ability and accuracy of representation. As such, they use a data driven design methodology that genuinely encompasses the motto *"let the data speak for itself"*. Unfortunately, such degree of flexibility and improved accuracy come at a cost of higher model complexity and huge computational resource requirements [30]. Some of these models follow a kernel-based learning. For example, the Gaussian process (GP), the support vector machine (SVM), ridged regression and kernel-PCA are few notable mention [39]. For instance, in the case of gaussian process regression, samples are taken from the multivariate distribution. Their plot against the input space looks a lot like the plot of a standard parametric function. However, these samples were drawn without any explicit assumption on the mathematical form or the number of parameters required to define the observed data.

Here, we would like to make it clear that parameterization of a model has nothing to do with it being probabilistic. Both parametric and non-parametric models can be made probabilistic. The parameterization affects how the model sees the given data and/or the kind of data it expects [37]. This will ultimately impacts its generalization and forecast accuracy. A better contextualization for the impact of parameterising a model can be illustrated by showing at how these models see a given observation. For instance, given a time series data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, a parametric model defines a probabilistic predictive model as

$$\begin{aligned} y_n &= f(x_n) + \epsilon \quad where \quad \epsilon \sim \mathcal{N}(0, \beta^2) \\ f(x_n) &= w^T x_n, \\ w &= [w_0, w_1, w_2, ..., w_d]^T, \\ x_n &= [1, x_n, x_n^2, ...x_n^d]^T \end{aligned} \tag{2.1}$$

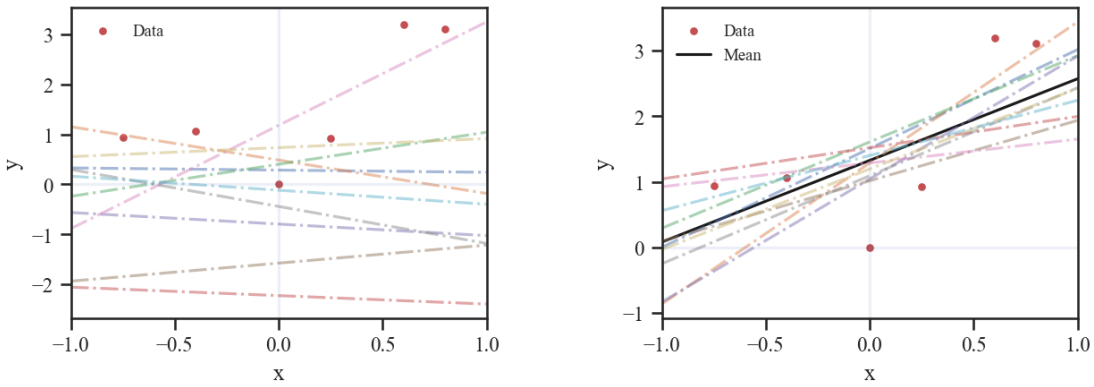
Where, x_n is the input point, w is the parameters of the model, ϵ is the perceived data variability, and y_n & $f(x_n)$ provide the model's actual and average output at x_n respectively. Alternatively, equation (2.1) can also be written assuming the parameters of the models as random variables as

$$\begin{aligned} y_n &= f(x_n) + \epsilon \quad where \quad \epsilon \sim \mathcal{N}(0, \beta^2) \\ f(x) &= Xw, \quad with \quad w \sim \mathcal{N}(\mu_w, \Sigma_w) \quad and \\ \mathrm{X} &= \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & ... & x_1^d \\ 1 & x_2 & x_2^2 & x_2^3 & ... & x_2^d \\ & & & ... & & \\ & & & ... & & \\ & & & ... & & \\ 1 & x_n & x_n^2 & x_n^3 & ... & x_n^d \end{bmatrix} \end{aligned} \tag{2.2}$$

Where, X is the design matrix containing input points in matrix, w is now a random parameter vector with its own mean vector μ_w and variance Σ_w. Here, the model assumes a specific parametric form for $f(x_n)$. As such, depending on the degree of complexity, a polynomial function of varying degree can be established. On contrary, the non-parametric gaussian process assumes a latent non-linear function $f(x_n)$ and defines a prior distribution over it as

$$\begin{aligned} y_n &= f(x_n) + \epsilon \quad with \\ \epsilon &\sim \mathcal{N}(0, \beta^2) \\ p(f(x) \mid \theta) &\sim \mathcal{GP}\Big(\mu(x), \Sigma(x_i, x_j \mid \theta)\Big) \end{aligned} \tag{2.3}$$

where $\mu(x)$ and $\Sigma(x, x)$ represent the mean function and covariance matrix of the multi-



(a) 1st order polynomial prior

(b) 1^{st} order polynomial posterior

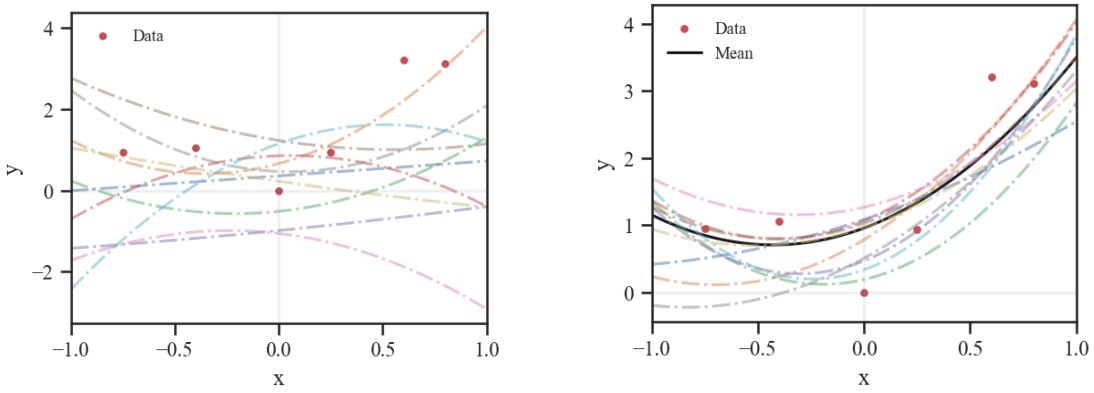
Figure 2.1: First order polynomial prior-post predictive distribution

variate distribution. Unlike the parametric model in equation (2.1), the Gaussian process in equation (2.3) assumes or specify no prior parametric form. Hence, its form is dictated by the data and covariance matrix defining the correlation between data points. As such, the objective is selecting potential functions that are consistent with the observed data. The target function $f(x)$ is given by the mean vector $\mu(x)$ and the variability between its corresponding data values is provided in the covariance matrix $\Sigma(x_i, x_j)$. For a probabilistic outcome, a prior distribution is assumed in both the parametric and gaussian process cases to train the model in a manner consistent with Bayesian reasoning. However, the approach and the impact it has, is different. In equation (2.1) and (2.2), the prior functional distribution is the result of the prior assumption on the type of the model or the specified number of parameters (w) and their corresponding distribution. This will define the nature of the model or the order of the polynomial. Hence, although probabilistic, all candidate functions for the data fitting will exhibit the same behavior but with varying parameter values as shown in the prior distribution of a 1^{st} order polynomial in Figure 2.1a. Contrarily, the Gaussian process in equation (2.3), defines a prior distribution over functions in a finite dimension equivalent to the range of observation. All kinds of functions differing in smoothness and wiggliness are presented as candidate functions as shown in the prior distribution of a Gaussian process in Figure 2.4a. Through Bayesian reasoning, only those functions consistent with the given data will be selected, averaged and presented as the final fit function. Further illustration of this difference is given in parametric and Gaussian process prior posterior distribution shown in Figures 2.1 - 2.4. Each figures display ten possible realizations from the prior and posterior distribution of 1^{st}, 2^{nd}, 3^{rd} order polynomial and Gaussian process fits for a sample data defined at

$x = \{-0.75, -0.4, 0, 0.25, 0.6, 0.8\}$ and generated according to

$$\begin{aligned} y &= 2sin(1.5\pi x^2) + x + x^2 \quad with, \\ \epsilon &\sim \mathcal{N}(0, x^2) \end{aligned} \tag{2.4}$$

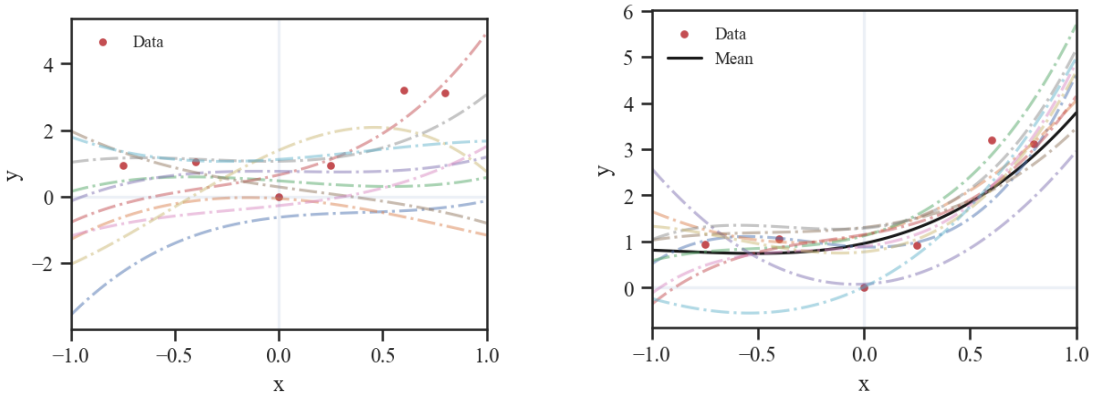
The impact of parameterization in constraining candidate functions for fitting can be



(a) 2^{nd} order polynomial prior (b) 2^{nd} order polynomial post

Figure 2.2: Second order polynomial prior-post predictive distribution

seen in the prior distribution of the polynomial models shown in Figures 2.1a - 2.3a. In Figure 2.1a prior distribution, only 1^{st} order functions are presented as candidates. The same analogy can be made in Figure 2.2a & 2.3a for quadratic and 3^{rd} order polynomials respectively. These prior assignments force the model to expect similar patterns in the observed data. Any data element that doesn't correspond to this pattern will not be represented accurately. This can be seen in the posterior distribution of the predicted function values shown in Figures 2.1b - 2.3b. The parametric form results in a computationally

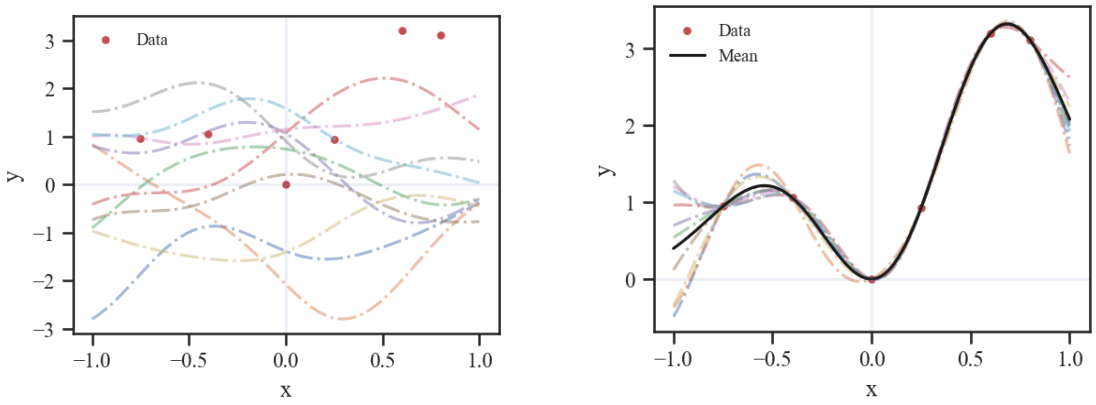


(a) 3^{rd} order polynomial prior (b) 3^{rd} order polynomial post

Figure 2.3: Third order polynomial prior-post predictive distribution

efficient model compared to the non-parametric form. The hope is that the prediction interval will be wide enough to compensate for any inaccuracies that might arise due to data variability, model specification or the approximation error introduced as a result of it. On the other hand, in Figure 2.4a, the Gaussian prior distribution presents complex functions as potential candidates without restrictions. Through an appropriate selection of kernel functions for the covariance matrix, the resulting distribution can made to accommodate the complex patterns (i.e linear, quadratic, periodic,...etc) observed in the data and expected to be seen the future. Hence, during training only those functions that best fit the

given data will be selected as shown in Figure 2.4b posterior distribution. The Gaussian process shows great flexibility in accommodating the given observation compared to the parametric forms as demonstrated in Figure 2.1 to Figure 2.4. However, this adaptability increase the complexity and computational cost of the underlying model. The reason being, the mathematical form and complexity of the generated functions is dictated by the covariance matrix. The dimension and contents of this matrix varies depending on the size of the data, the type, kind and number of kernel functions used. As a result of these interdependence, the covariance matrix is synonymously called the kernel matrix. And we will be using these two names interchangeably in the rest of this thesis.



(a) Gaussian process prior

(b) Gaussian process post

Figure 2.4: Gaussian process prior-post predictive distribution

One feature of data that makes prediction possible is the correlation between observed data points. Without local or global interdependence within the data, prediction will be impossible. The observation will not contain any valuable information regarding what comes next. Consequently, the recorded data doesn't provide a valuable insight about the probable future expectations other than being considered as a random noise. Some other time, even in the presence of a valid correlation, the relationship will be too complex to comprehend for some models, like auto regressive models (i.e AR,MA,ARIMA,...etc). Fortunately, for Gaussian process model the concept of kernels creates a mathematical convenience which enables it to uncover the hidden associations between data points. The models employ kernel functions as a measure of similarity between data points regardless of the extent or complexity of the correlation. In return, this similarity index will be used to provide an appropriate function which is consistent with the observed data and agrees with the observed similarity [40]. However, searching and selecting suitable kernel functions for the covariance matrix is not an easy task and mostly it is challenging [40].

The basic concept behind kernel based learning is understanding the correlation between data points. This help us understand the structure of the data. Models like the Gaussian process whose inference is based on the Bayesian framework will use it to mold a prior distribution on the given data [35, 40]. The model employs various kernels to establish the trend, local and global similarities (i.e smoothness and periodicity) between observations [41, 42]. The extent of these co-variances are regulated by the parameters defining each kernel functions. These parameters determines the degree of forecast horizon at which the model could extrapolate. For example, the prediction horizon for a Gaussian process with a squared exponential kernel function is proportional to the length (l) parameter. For a prediction horizon larger than this length, the model prediction will revert back to the mean value. Consequently, how far to the future a model can see is dependent on the kernel type and its parameter values. These limitations can easily be

rectified through a careful selection of kernels and parameter optimisation in such a way that the performance of the model is optimal.

The biggest challenge in the implementation of Gaussian process is the dimension of the kernel matrix. The Gaussian process model defines NxN covariance matrix (Gram matrix) employing different combinations of base kernel functions, where N signify the size of the data [43]. The Gaussian process needs the inverse of the covariance matrix for parameter optimisation and posterior predictive distribution. A procedure that requires an operation cubic to the number of data instances $\mathcal{O}(N^3)$ [39, 44]. The computational of the model is further exacerbated by the number of kernel functions used for data fitting which increases the number of hyper-parameter during optimisation [42]. In addition to the computational burden, the persistent need to keep the data for training and prediction puts a higher memory requirement. As a result the direct implementation of kernel-based learning on large data has been challenging [39]. This is the very fact that forbade the scalability of Gaussian process to big data domain and constrain their application areas only to smaller data domain. Later, we will return to the topic of kernels and their impact on the predictive performance of the model in detail in section 2.4.

2.3 Gaussian process model limitations

Nowadays non-parametric predictive models are gaining momentum. Among them, probably the most under represented and utilized regression algorithm in the areas of electricity load forecasting is the Gaussian process (GP) [21]. GP have been used extensively in the areas of classification and regression due to their nonparametric nature and predictive performance [45]. Time series analysis and prediction [46, 47], stream flow prediction for water resource management [48], generated energy forecast for solar [49], wind [50], state of health prediction for battery [51], tourism [52] and demand forecasting [21, 34, 35, 49, 53, 54, 55], are few notable mentions. Their non-parametric form allowed them to have the flexibility to fit complex function in infinite dimension in a way that is interpretable and puts a measure of cost on overfitting. Additionally, a closed form solution for inference and hyperparameter optimization make the models more interesting. However, they don't always explain the observed data. This is due to the unrealistic assumption that the prior distribution of the observed data is Gaussian-alike [56].

One of the desirable feature of GP is its ability to provide forecast distributions as opposed to point estimates. Energy companies use forecast distribution as a basis for making decision on unit commitment, energy price fixing and distributed energy resource integration [54]. The recent trend toward smart grid resulted in huge volume of data. Applying Gaussian process regression on such data incurs a considerable memory requirement and computational cost. While the prediction distribution offered by the model is attractive, the computational resource requirement remains higher. The success and limitation of Gaussian processes can be attributed to two factors. The Bayesian framework that is used for parameter learning & inference, and the adapted kernel based learning approach for measuring the correlation between data points.

2.3.1 Kernels

In the words of Arthur Samuel, the pioneer in the fields of AI, the idea behind machine learning is to give computer models the ability to learn from data without explicitly being programmed to do so. Learning imply the capacity to find a valid relationship within the data, identifying redundant features, differentiate regular and irregular occurrences towards building a model for inference and generalization. Such pattern identification and data exploration will not only help in building a possible data generation model or an

approximation of it, but it also improve the expectation of the model about the possible data that is expected from the source. Among the different algorithms used in pattern analysis, kernel based learning is one of them. Kernel functions were first used for pattern analysis in SVM classifications tasks. Their implementation has improved the flexibility of classification algorithms and gave a computational advantage over linear classifiers in solving non-linear functions or decision boundary problems [57]. Over the years, research in the fields of kernels and pattern discovery has seen them in various application areas such as regression, classification, correlation, principal component analysis, clustering and many more. In Gaussian process, these functions are mainly used to identify the similarity over all pairs of observed points. As part of the learning process, mapping the correlation remained the responsibility of the specific kernel function used, whereas the inference and prediction is left to the Bayesian framework. Hence, they both have a part to play in determining the predictive performance the underlying model [58].

The Gaussian process defines a prior distribution $p(f(x) \mid \theta) \sim \mathcal{GP}(\mu(x), \Sigma(x_i, x_j \mid \theta, k))$ over functions using different combinations of kernels k. A mean centered prior definition which assumes $\mu(x) = 0$ is preferred for mathematical convenience [59]. As such, the breadth and depth of the prior distribution is entirely dictated by the covariance matrix Σ, the kernel function k and values of the its parameters θ. Being Bayesian demands the consideration of prior knowledge or belief. In the absence of data, they represent the ground in which we base our decision on. Meaning that how they are defined, whether it is random or it is based on prior expertise determines the accuracy and computational efficiency of the underlying model. For instance, in univariate Gaussian, uninformative and broad prior definition (i.e Uniform distribution) could drag the computational time required for parameter evaluation. A narrow assignment can be too restrictive. As such, it can constrain the parameter space which can mask the estimation of the right values. The same can be said in the multivariate distribution. For example, in Gaussian process regression, the Bayesian framework assign probabilities to every sampled functions from the distribution. This measures the likelihood of the given function is representing the observed data. A wide prior assignment expands the function evaluation space, thereby forcing the estimation to take a longer time. The opposite can also be said for a narrow prior assignment. This will constrains the space of possible candidate functions. As such, the true function representing the observed data might not be available. Consequently, a balanced prior assignment that is based on prior expertise in relation to the problem at hand is fundamental for the accurate modeling and representation of the observed data. Such assignments are regulated by the choice of kernels functions and their corresponding parameters.

Researchers have pursued a parametric and non-parametric approach to kernel function design. The non-parametric method encapsulates the unconstrained and data-driven approach to kernel design [60]. Especially, the absence of parametrization in its implementation has circumvented the need for specifying complex covariance functions. Consequently, simplifying kernel-based learning [39, 61]. For example, the hierarchical Bayes [61] follows a non-parametric approach for a data-driven kernel design. The approach utilized a cascaded EM and Nyström algorithms for the estimation and covariance matrix generalization to new features. Another notable mention is the Bayesian non-parametric kernel learning BaNK [39] algorithm. BaNK provides a robust generative model that is scalable to large data based on gaussian mixtures for a data-driven kernel design. Such probabilistic based approach allowed a large class of kernels to be estimated at a time. Hence, providing a superior performance for a regression and classification tasks. Nevertheless, the hyper-parameter optimization follows MCMC sampling which incurs a time constraints for a posterior convergence. On the other hand, the parametric approach offers a fixed batches of basis functions. This will restrict possible kernel explorations and

bounds pattern learning to the subsets of few priorly chosen kernel functions [39]. Moreover, the choice of a candidate basis function from the set requires deep insight about the data and descriptions of the kernel function [40]. However, coupled with a good search algorithm, these primitive function can provide an acceptable predictive performance. As a result, they have been used extensively in kernel-based learning. The compositional kernel search methods such as the structure discovery [40] and dynamic kernel search [42] are few notable mentions.

The limiting factors in the implementation of a search algorithm for suitable kernel functions can be attributed to the nature of the inference applied and the size of the data. Gaussian inference requires operations cubic to the size of the data. As such, any attempt on the algorithmic kernel selection would requires a continuous training and evaluation of the model. This will ultimately affect the memory and time efficiency of the model. Consequently, in addition to the exploration mechanism, the implementation should address the limitation of the underlying model. As such, a faster and more practical approach in model building that is scalable to large data must be followed. The computational burden attributed to the size of the data prohibits the implementation of the exact gaussian inference in big data [60]. Approximation methods and minimizing the number of training instances are the two widely used alternatives for scalability and computational efficiency. For example, ensembles models [55, 62], gaussian scalability through kernel manipulation [43, 63], localized regression by splitting the observational space [44, 64, 65], gaussian approximation through variational and MCMC sampling [66, 67, 68, 69, 71] are few notable mentions. Localisation and minimization of the observational space by retaining and discarding instances of the training data, reduces the dimensionality of the kernel matrix. Thereby improving the model performance and making the search algorithm feasible. However, this technique inadvertently affect the inferential capacity of the model and eclipses the hidden patterns that could be learned.

On the other hand, when minimizing the span the observation is not an option, variational and MCMC approximation methods [66, 67, 68, 69, 71] have been put forth as a viable alternatives. These methods have addressed the kernel dimensionality crisis and the Bayesian posterior computational bottleneck. This has dramatically improved the computational efficiency and scalability of the model. In doing so, opened the door for an acceptable kernel search implementation [42]. Nevertheless, the variational inference requires a lot of computation and iterations for the ELBO convergence. As such, running the search algorithm although possible, consumes a considerable mount of time. The same reasoning can be extended to models based on MCMC approximation in regards to the effort needed for a posterior convergence.

The computational hurdles being the main issue, the success of kernel based learning is also dependent on the users ability to select an appropriate kernel. Various approaches have been suggested to facilitate the search for suitable kernels for GP. For example, the structure discovery [40] and the dynamic kernel [42] are notable mentions for finding possible kernel combinations for the given observation. The structure discovery algorithm in [40] employ a greedy search approach using the marginal likelihood as a criterion to pick the highest scoring kernel. This kernel again is used as a basis to find other mixtures. For a Gaussian process that consumes huge computational resources, this is a brilliant approach. The greedy search approach, where we take the local best scoring kernel can shorten the search time, However, it can also eclipse other combinations that best fit the data. Whereas the algorithm in [42], performs an exhaustive search using the mean squared error (MSE) as criterion for evaluating suitable kernel combinations and forwards the combinations with lowest MSE as the optimal kernels. Although these approaches improve the predictive accuracy of the model, the number of kernel functions and complexity of their mixture further exacerbate the computational burden.

2.3.2 Bayesian framework

In the case of the framework, the Bayesian is what gives the GP its simplistic inference. However, it is also responsible for constraining the application of the model in areas of big data and in the areas where the normal Gaussian assumption doesn't hold [56, 71]. This is due to the inherent computational constraints in evaluating the posterior distribution. The power of Bayesian inference rests on its ability to accommodate the contribution of model parameters by averaging over all likely values they could take under its posterior distribution. To that end, the inference needs the inverse of the kernel matrix $(\Sigma_{xx} + \sigma^2 I_n)^{-1}$ for parameter optimisation and posterior computation. However, the inversion of a matrix is something which is not computationally feasible, even under the best of terms (i.e small matrices), let alone for a kernel matrix whose size is equivalent to the size of the training data. This fact has reduced the computational efficiency and created scalability issues to large data. Furthermore, a data distribution other than the Gaussian likelihood assumption, often renders the posterior computation intractable [70, 71]. In an effort to address issues related to the kernel dimensions, different approaches have been suggested. For instance, clustering the training data locally and training an ensemble of Gaussian processes is presented as an alternative [64]. They adopted a weight assignment that follows a probabilistic approach where each model is assigned a weight based on the average likelihood of containing the predicted point. Then, the forecast mean is evaluated as a weighted average of the response of each ensemble models. However, this approach effectively transform the non-linear relationship between data points and linearize the mean function around the forecast points . As such, for data exhibiting complex pattern (i.e periodicity) the approach offers a sub-optimal prediction accuracy. Another approach to solve computational complexity in GP is to probabilistically divide the observational space using gating networks and fit a specific model for each sub region [65]. This has the advantage of significantly reducing the dimension of the kernel matrix where the computational effort is cubic only to the number of data points contained within the region [65].

Approximation based estimation offers a viable alternative for a scalable Gaussian process. Any model that ever existed is an approximation and a simplification to the real system that generated the data and the uncertainty that affected it. The advancements in the areas of computing power, data mining, and optimization has reduced the dependency on approximate mathematics for solving intractable equation just for the sake of convenience [72]. These days computers have become far better in solving intractable equation numerically than they were years ago. Hence, the application of numerical methods is becoming more prevalent in solving large scale problems than a standard closed form equation. Especially, the computational power is advancing to the point where the choice between optimisation algorithms is becoming irrelevant [37]. However, having the power for solving equation doesn't signify there won't be an approximation error.

The notion behind approximation is that the overall dynamics of the system can be replaced by a model with a smaller set of parameters. This results in a more computationally efficient and close resembling counterpart. Sampling based estimation methods (i.e MCMC) employ random sampling to provide an approximate solution to a deterministic intractable equation. Inherently these methods are not regarded as an optimization algorithms. In a sense, they don't provide estimate that would maximized the likelihood of the given cost function. But rather, they provide a potential predictive distribution for those values that would maximize the likelihood of the underlying event. They approximate the posterior, first by drawing a random sample from the initial distribution. Then, iteratively move toward the desired distribution by applying a stochastic transition operator until convergence [71]. This method offers simplicity in terms of implementation. However, its accuracy and efficiency are dependent on the number of iterations

and transitions required. The variational counterpart, on the other hand, presents the question of approximation as an optimization problem. Hence, it defines a parameterized surrogate distribution and optimize it until the divergence from the true distribution is minimal [66, 71, 73]. It applies the KL-divergence as a metrics to quantify this difference. The intractability of the Bayesian posterior is due to the difficulties of estimating the normalization constant. The KL-divergence is insensitive to constants. As such, its inclusion in minimization renders the approximation process insensitive to the effect of the normalization constant. This has greatly simplified the computation of a surrogate distribution to the true posterior distribution.

The sparse variational GP uses variational inference and sparsity to address the challenges of posterior intractability and input dimensionality. The approach approximate the posterior through variational inference utilizing few randomly sampled points from the input space [66]. These points and the associated values are called the inducing points and inducing random variables respectively. This technique effectively reduced the kernel dimension and improved the computational effort. Hence, it is regarded as the derivative of GP that truly scaled the model to the big data domain. Its predictive efficiency has been demonstrated in the areas of navigation [73], optimal sensor location for communication [74], classification [67] and regression [68, 69]. Nevertheless, the inability to determine the number and location of these inducing points limits the forecast accuracy of the model.

Predominantly, issues of scalability has been addressed either through approximation or down sampling the data. Both approaches have their own advantages and limitations. Down sampling or discarding, relies on taking few prior observation at a time with a walk forward forecast update for a continuous prediction. Such approach usually employ a gaussian kernel to define the correlation between the observed and the forecast points. Consequently, it gives high emphasis to points closer to the forecast horizon. Inevitably localizing the model and forcing it to disregard points that are further away and those which can affect the forecast point periodically. Although discarding part of the data ensures the adaptation and scalability of the model, it also results in a loss of valuable information. Thereby, minimizing the models ability to make sense of complex interactions between recorded observations. On the other hand, Bayesian backed approximation methods have dramatically improved the computation efficiency and scalability the models, though with a reduced accuracy. Such approximations are widely carried out either through sampling (i.e MCMC) or applying variational inference [71]. In the absence of such estimation, exact gaussian inference on large data is bound to increases the dimension of the kernel matrix. Thereby increasing the computational burden. As a result, a reasonable trade off is made with dimensional reduction for a fast and efficient model through approximation at the expense of exact solution. This approach has reduced the application areas and limited its implementation on big data domain [43]. However, a great stride has been taken to improve its predictive accuracy. Ensembles models [62], aggregates [55], kernel manipulation for scalability [43], and localized regression by splitting the observational space [75], are few notable mentions. Despite the different approaches followed in constructing the predictive model or the algorithm used to optimize the hyperparameters, the success of GP for large data has relied on minimizing the number of observation in training set. As a result, the adaptation and/or scalability of the model and long term prediction rely on data reduction through down sampling.

2.3.3 Prediction interval width estimation

In predictive analytics, model building for a specific application and making a forecast for future outcomes may not always be enough. In some areas like banking, finance and

energy sectors, more is demanded from a predictive model than just a prediction. The model is expected to provide a measure of how confident it is about that specific forecast. Such information is invaluable in risk management, mitigation and decision making.

In section 2.1, we have stated how the explicit consideration of noise in model building empower the model to have more than just point forecasting. Once the presence of uncertainty is recognized, then the a stochastic approach to parameter assignment and all variables affecting system behavior as random is logical. The sources of these uncertainties can easily be the parameters of the model whose values are unknown, inadequate modeling due to lack of exact mathematical understanding of the underlying process, model approximation, numerical approximation during optimisation, data variability during measurement so and so forth. The characterization of uncertainties are as wide as the errors we are bound to make when modeling. However, their categorization as internal to the system (Epistemic) and external to the system (Aleatoric) will generalize them better. Such identification helps the model to understand and quantify the uncertainty. Regardless, the most important part beyond their recognition is the question of how to quantify them and minimize their subsequent impact on prediction. It has been an accepted norm that an informed decision should take into account the possible uncertainty that could be expected and the decision should be made considering this [76]. Although, different methodologies have been followed in their estimation, an intuitive way of presenting the range of possible values or the degree of belief in estimated parameter has been through confidence interval.

Interval estimation uses different methodology depending on the nature of the data and the modeling technique applied. For instance, in linear regression and Bayesian model, the residual error (noise) is assumed to follow a normal distribution. In case of a smaller sample size, the t-distribution is applied as an alternative sample distribution for interval estimation. However, there are occasions where the assumptions of normality doesn't hold. The data generation process may be too complex, follows unknown distribution or it might just follow an asymmetric distribution. In such cases, an approximate distribution, bootstrapping and quantile methods are usually employed. On the other hand, in deep learning models, a data driven(non-parametric) approach to variance estimation and a bimodal distribution for parameter sampling are used to estimate this interval. Regardless of the techniques applied, the most important aspect of interval estimation is its precision. And, how precise it is depends on many factor. The estimation can easily be affected by the inadequacy of the modeling approach, quality of the data or failure in accounting the sources of uncertainty mentioned above. Without addressing these scenarios adequately, even if the model provided a confidence interval, the information conveyed will not be accurate and estimation will not be useful other than providing a false sense of security. Beside the precision, how wide should it be ?, is yet another question worth asking. A naive predictive model can account for all sources of uncertainty by predicting a wider confidence interval that delivers a 100% coverage probability of the forecasted points. Inherently, it is logical to say a minimum confidence interval with maximum coverage is a good quality to have for a predictive model. Meeting this objective has been a challenge for most predictive models and that is the very reason we are mentioning it here as an additional limitation to the gaussian process.

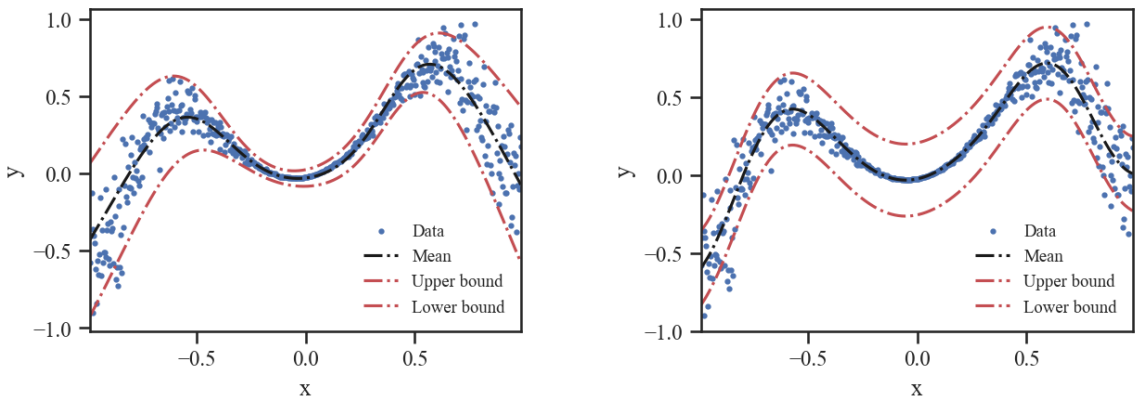
In section 2.3, we have tried to list some of the limitation of the Gaussian process. These limitations have more to do with the accuracy of the mean trajectory and computational efficiency of the model. This limitations can be minimized through a proper selection of kernels, applying the sparse and variational approximations. Its confidence interval estimation, however, is dependent on the distance between observation and their uncertainty. This fact is self-evident in the values of the covariance matrix elements. Assuming a squared exponential kernel is selected, the elements of the covariance matrix

(Σ_{ij}) is given a value corresponding to the distance between observations x_i and x_j

$$\Sigma_{ij} = \alpha^2 * exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \sigma_{ij}^2 \tag{2.5}$$

where σ_{ij}^2 is the assumed data variability. Meaning that a smaller distance between observation results in a higher correlation. In other words, the more data we have the less ambiguity in model prediction which results in a minimum confidence interval. Additionally to account for any modeling errors, usually the 95% coverage probability is taken as a design parameter for the upper and lower bound interval estimation. However, there are no frameworks to check the optimality of the confidence interval and whether or not the forecasted points actually lie within the interval. This is not the limitations of a Gaussian process alone, but also a shortcoming shared by most predictive models.

The primary goal of model building has been to construct a data generation model that can closely resemble the observed data, an almost perfect match between the inputs to the outputs [77]. As a result, the methods of fitting and training is solely devoted to ensuring the accuracy of the target trajectory. For example, the Gaussian process models are trained using the maximum likelihood as a criterion. Here, the aim is to maximize the likelihood of generating the observed data without considering the prediction interval or its coverage probability. Hence, for objective assessment and quantification of the optimality of the confidence interval, predictive models should be trained in a manner that takes into account not only the accuracy of mean trajectory but also the interval width and coverage probability. Figure 2.5 describes the difference in the confidence in-



(a) Coverage with MPIW and PICP

(b) Coverage with Maximum Likelihood

Figure 2.5: Effect of estimation approaches on the coverage probability and interval width

terval generation between two models trained on different algorithms. Figure 2.5a shows the confidence interval estimation for model trained on an algorithm that tries to optimise the interval width and coverage probability while Figure 2.5b shows a model trained using maximum likelihood criterion. In chapter 4, we will have a lot to say about algorithms focused on the optimality of the confidence interval, but for the time being, from Figure 2.5a, it can easily be verified that an algorithm that takes into account interval estimation delivers a smaller interval with maximum coverage compared to other alternative methods. In Gaussian process, interval estimation that focus on optimising the width and coverage probability of the forecast horizon is not practically feasible. On contrary, deep learning models present an opportunity to try new algorithms and provides a good platform for testing uncertainty quantification and bound estimation. Furthermore, unlike the Gaussian process models, deep learning models do not have a scalability issue to large data. By widening their parameter space and accumulating as much parameters as need commensurate with the given data, they try to provide a fitting function. Though,

often times running into the risk of overfitting and over-parameterized models. However, the evolution of big data and the flexibility of these models, has seen a wide penetration of deep learning models in demand supply prediction [78]. However, like the Gaussian process, they have their own limitations. One of the major challenge in deep learning models is point estimation and their absolute certainty in the accuracy of their prediction. Especially, in areas like energy demand and supply prediction, where wrong prediction can cause a devastating impact on the stability of the grid, over confidence is dangerous. However, researches in the areas of deep probabilistic models such as Bayesian neural net (BNN) have enabled these models to handle data variability so that their prediction can be supported by a margin of uncertainty. In chapter 4, we will revisit uncertainty quantification in deep learning models in great detail and introduce a new algorithm for interval estimation.

2.4 Kernels in gaussian process

The word kernel carry different meaning depending on the area of specialization. In linear algebra, it is a reference to the nullspace (linear subspace) containing all vectors $\vec{v}$ that provides a solution to the homogeneous systems whose dynamics are described by the equation $A\vec{v} = 0$. In computer science, it is often associated with the core program responsible for facilitating the communication between the hardware and the operating system, and it could also refers to the smoothing function in kernel density estimation problems. However, in here kernel refers to all functions in the reproducing kernel hilbert space (RKHS) which are used as a measure of similarity. The Hilbert space is an abstract mathematical space created by the inner products of random vectors which allows efficient computation to be carried out irrespective of the dimension of the space. As such, the role of a kernel function in general is returning a similarity index between any two pairs of data points [79]. The subject of kernel is vast and so is their contribution to the areas machine learning. For instance, one of the difficult task in classification is classifying a non-linear separable data. Transforming it into a linearly separable data requires the introduction of auxiliary features or new dimensions to the existing data. This process is accomplished by mapping the data onto a higher dimensional space using a non-linear function $\phi(\cdot)$. This procedure can get computationally intensive real easy. It will be more exponentially demanding when the size of data and number of features considered are substantial, as we are expected to apply the mapping function $\phi(\cdot)$ on each element and later on perform computations of the resulting features. Kernels, on the other hand, provides an efficient mechanism of computing the inner product that is required for setting the decision boundary without the need to map the data onto a higher dimension. This makes makes the task of data mapping computationally feasible. The type and application of kernels is broad and it won't be possible to review them all here. However, an in depth analysis on kernels and their corresponding features space in the form of a reproducing kernel map can be found in [57, 80, 81]. As kernels are the information block of the Gaussian process, for the sake of generality, before going deeper into the mathematical details and analytical solutions of the model, it will be best to explore and familiarize ourselves with some of the kernels that are used in Gaussian process for an interpretable predictive distributions.

A kernel function $k(x, z)$ quantify the similarity between any two given vectors $x = [x_1, x_2, \cdot\cdot, x_m]^T$, $z = [z_1, z_2, \cdot\cdot, z_m]^T \epsilon \mathcal{R}^m$, through their the dot product (inner product)

$$k(x, z) = x^T z$$

or, equivalently

$$k(x,z) = \sum_{i=1}^{m} x_i z_i \tag{2.6}$$

$$k(x,z) = \frac{x \cdot z}{\|x\|\|z\|}$$

From equation (2.6), it is simple to deduce that $k(x,z)$ is symmetric $k(x,z) = k(z,x)$ and $\forall\ z\ \epsilon\ \mathcal{R}^m$, $z \neq x$, $k(x,x) > k(x,y)$, meaning that a vector is more similar to itself than another vector. Equation (2.6) can also be extended to any set $\phi(\cdot)$ that acts as feature representation of the vectors. In classification tasks, the function $\phi(x)$ represents the mapping function that transform a non-linear relationship to a linear in higher dimension. In regression, given a data set $\{(x_i, z_i)\}_{i=1}^m$, its purpose is to creates a number of features $\{(\phi(x_i), z_i)\}_{i=1}^m$ which could possibility enhance the predictive performance of the model.

Assuming there is such mapping function $\phi(\cdot)$ that transform the vectors to a higher dimension $\phi(\cdot) : \mathcal{R}^m \mapsto \mathcal{R}^n$ where $n > m$, the kernel $k(x,z)$ can also be defined as

$$\begin{aligned} k(x,z) &= \phi^{'}(x)\phi(z) \quad or \\ k(x,z) &= \frac{\phi(x)\phi(z)}{\|\phi(x)\|\|\phi(z)\|} \end{aligned} \tag{2.7}$$

The best feature of kernels is that, even if $\phi(\cdot)$ is regarded as a set containing the features of the given data, there is no need to create the multivariate data representation of it, as it can be evaluated automatically using the predefined kernels. In equation (2.6) & (2.7), $k(x,z)$ follows a dot product and hence, the result is a singular value. However, applying $k(\cdot,\cdot)$ on each element of vector $x\ \epsilon\ \mathcal{R}^n$ creates a similarity matrix $\mathrm{K}\ \epsilon\ \mathcal{R}^{nxn}$ containing all possible inner products

$$\mathrm{K} = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdot & \cdot & \cdot & \cdot & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \cdot & \cdot & \cdot & \cdot & k(x_2,x_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1},x_1) & k(x_{n-1},x_2) & \cdot & \cdot & \cdot\cdot & & k(x_{n-1},x_n) \\ k(x_n,x_1) & k(x_n,x_2) & \cdot & \cdot & \cdot & \cdot & k(x_n,x_n) \end{bmatrix} \tag{2.8}$$

This matrix is called the kernel matrix K or the Gram matrix. **Note:** we will be using the capital letter K and Σ interchangeably to refer to the covariance or kernel matrix. The representation of kernel as a matrix instead of a function will simplify the analysis. Such transformation will allow us to bring the techniques of linear algebra to the table without distorting the fundamental characteristics of the kernel function [81, 82]. The kernel matrix (K) is symmetric (i.e $K = K^T$) and positive semi-definite if and only if $\forall z\ \epsilon\ \mathcal{R}^n$, $z^T\mathrm{K}z \geq 0$.

Proposition 2.4.1. *A matrix K is positive semi-definite if and only if $K = L^{'}L$ for some matrix L*

Proof. Suppose $A = L^{'}L$, then for any vector λ, we have $\lambda^{'}A\lambda = \lambda^{'}L^{'}L\lambda \Rightarrow \|L\lambda\|^2 \geq 0$ □

The kernel matrix and its inverse are frequently used in parameter optimisation, sampling and estimating the multivariate posteriors distribution. Despite its mathematical convenience, the matrix representation doesn't spare the computational and resources requirement of the underlying process from being higher. The symmetricity and positive

semi-definiteness properties of the matrix can be used to lessen the burden. The symmetric property ensures that the matrix can be factorized into a more computationally efficient form of representation. To that end, a number of numerical methods have been suggested. Matrix decomposition methods, like Eigen and Cholesky, apply factorization techniques in order to provide a closed-form analytical solutions to a matrix with a less resource requirement. The Cholesky decomposition factorize the kernel matrix K as a product of an upper and lower triangular matrix (i.e $K = L^{'}L$). This approach has enabled the inverse and determinant computation to be performed more efficiently. Especially, the computational gain will be apparent when inverting large matrices. The contribution of Cholesky method goes beyond linear algebra and sampling Gaussian distribution. It is also applied in a stochastic deep learning optimisation as a reparameterization trick which transform a random variable into a differential variable so that its gradients can flow through the network during the back propagation step. The are also other numerical methods such as the Nyström method that rely on sampling and recursive estimation in order to provides a low rank approximation to the kernel matrix. Consequently, the Cholesky and Nyström methods are considered alternative numerical methods that are frequently used in practice to enhance computational efficiency [82].

The positive definiteness property of a matrix, geometrically creates a conic vector space that is more amicable to quadratic optimisation algorithms. In the quadratic or convex optimisation problem the aim is to provide a solution to a multivariate function $f(x) = \frac{1}{2}x^T Kx - Bx + c$. The search for the vector x^* that minimize $f(x)$ will be easier to work with when K is positive definite guaranteeing that the function $f(x)$ is a convex function, its associated hessian matrix ((i.e $f^{''}(x)$) positive and the solution x^* a global minimum. Unfortunately, the gaussian process objective function for parameter optimisation is a non-convex function. As such, global minimum value is not guaranteed. In the coming section, we will formally define and frame the parameter optimisation in the gaussian process as non-convex optimisation problem.

The characteristics mentioned above have more to do with how to realize efficient computation when the size of the data or the dimension of the kernel matrix is large. The positive semi-definiteness of the kernel matrix doesn't imply invertibility. There are moments when the kernel matrix becomes singular (i.e $det(K) = 0$) even for smaller data and evaluating its inverse may not be possible. For instance, identical data points could result in similar values providing dependent eigen vectors which makes the matrix ill-conditioned. In practice, to avoid this singularity a sort of regularization term (i.e an infinitesimal number) is added to the diagonal part of the existing matrix (i.e $K = K + \alpha I$, where α is a very small constant number), as a means of ensuring the stability of matrix inversion. This infinitesimal number is sometimes called a noise term or a jitter.

Any function can be regard as a kernel function provided it results in a similarity matrix that is symmetric and positive semi-definite.

Proposition 2.4.2. *For any two kernel functions k_1 and k_2 that fulfill proposition 2.4.1, then the following are also kernels*

1. $K = \alpha_1 k_1 + \alpha_2 k_2$ *for any* $\alpha_1, \alpha_2 > 0$
2. $K = \alpha_1 k_1 * \alpha_2 k_2$ *for any* $\alpha_1, \alpha_2 > 0$

Proof. Given k_1 and k_2 are positive semi-definite and for any random vector $z \epsilon \mathcal{R}^n$,
If $z^T k_1 z \geq 0$ and $z^T k_2 z \geq 0$, then summing both $z^T(k_1 + k_2)z \geq 0$, and
$\forall \alpha_1, \alpha_2 > 0$ If $z^T(k_1 + k_2)z \geq 0 \Rightarrow z^T(\alpha_1 k_1 + \alpha_2 k_2)z \geq 0$
If k_1 and k_2 are positive semi-definite, then the element wise multiplication $k_{1ij} * k_{2ji}$ results in a positive semi-definite kernel $K = k_1 * k_2$. An elaborate proof for the positive-definiteness of the Hadamard product $(k_1 \circ k_2)$ is given in [80]. □

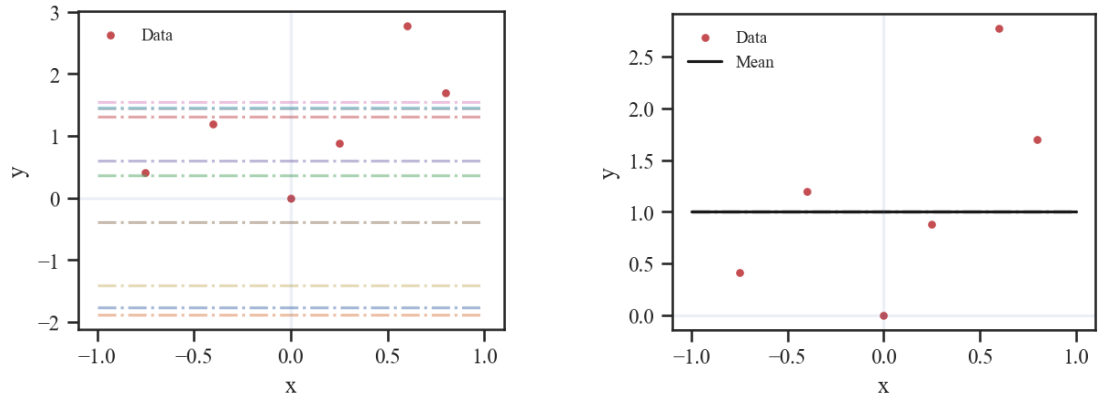
Proposition 2.4.2 lays a theoretical foundation for the construction of new kernels out of the combination of the existing ones. On any function that fulfills the characteristics of a kernel, the sum and product rule of proposition 2.4.2 can be used to construct new kernels.

2.4.1 Constant kernel

The constant kernel defines a matrix of all ones scaled by some parameter σ. Each elements of the matrix K has the same value $k_{ij}(x_i, x_j = \sigma^2)$, depending on the value of the parameter.

$$\mathrm{K} = \sigma^2 \begin{bmatrix} 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{\sigma}{\sqrt{n}} & \cdot & \cdot & \frac{\sigma}{\sqrt{n}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\sigma}{\sqrt{n}} & \cdot & \cdot & \frac{\sigma}{\sqrt{n}} \end{bmatrix}^T * \begin{bmatrix} \frac{\sigma}{\sqrt{n}} & \cdot & \cdot & \frac{\sigma}{\sqrt{n}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\sigma}{\sqrt{n}} & \cdot & \cdot & \frac{\sigma}{\sqrt{n}} \end{bmatrix} = V^T V \tag{2.9}$$

It is a symmetric and positive semi-definite matrix as given in proposition 2.4.1. It can easily be proved for any random vector $z \epsilon \mathcal{R}^n$, it can be decomposed and rewritten as $z^T K z \Rightarrow z^T V^T V z \Rightarrow \|Vz\|^2 \geq 0$. As per the proposition 2.4.2, it is mainly used to create and modify other kernels. In practice, it is employed to shift the posterior mean as an additive kernel or scale it as product kernel. Hence, applying the constant kernel as the covariance function only produces constant line functions. This can be seen from few sample functions drawn from GP prior with the constant covariance function for the data given in equation (2.4) as shown in Figure 2.6a.



(a) Constant kernel GP prior

(b) Constant kernel GP posterior

Figure 2.6: GP with constant kernel predictive distribution

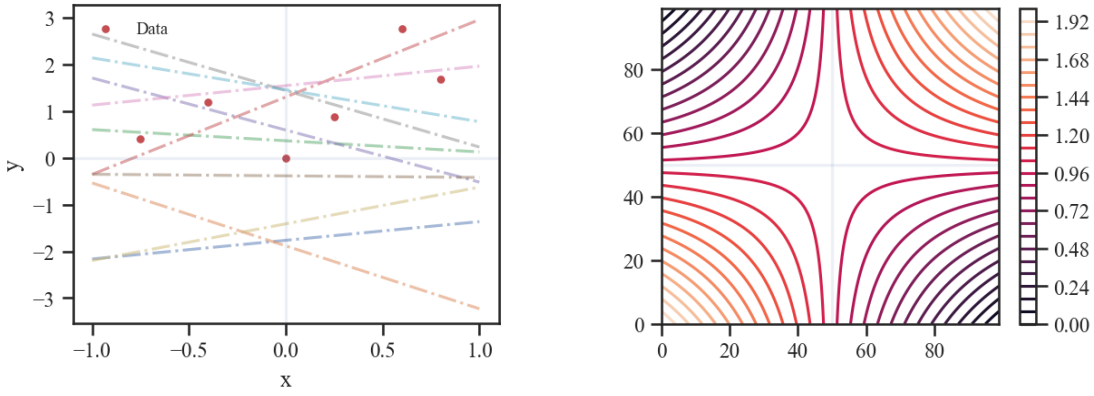
2.4.2 Linear kernel

The linear kernel is a non-stationary kernel that defines a dot product between the elements of two vectors x and z.

$$k_d(x, z) = \lambda(x^T \cdot z) \tag{2.10}$$

Because of that sometimes it is referred as the dot-product kernel. The parameter λ is sometimes replaced with a constant kernel. According to the second item of proposition 2.4.2, the linear kernel can also be given as a product of the constant kernel and the dot-product kernel(i.e $K = K_c(x, z) * k_d(x, z)$) without the parameter λ. In classification, it is mostly used when the existing data has enough features or exhibit higher dimensionality or when the diversity and type of data allows linear separation. In regression, they

are employed to fit first order linear models and also as an additive and product kernel to model upward and downward trends in a data. The covariance map and few sampled functions drawn from a gaussian process prior with a linear kernel for the data given in equation (2.4) is shown in Figure 2.7b and Figure 2.7a respectively. From Figure 2.7a, it can be generalized that a gaussian prior with a linear kernel as a covariance function only produces straight lines or models linear functions.



(a) Linear kernel GP prior

(b) Linear kernel covariance map

Figure 2.7: GP with linear kernel prior distribution

2.4.3 Polynomial kernel

Polynomial kernels are a generalization to the linear kernels. They are a valid non-stationary covariance kernel functions. Their validity can easily be proved by applying the second item of proposition 2.4.2 and representing it as a product of p number of linear kernels as shown in equation (2.11). They are mostly used when the data exhibit non-linear variability and the addition of extra features or dimensions are thought to improve the performance of the data generation model. These kernels allows complex features to be considered during model development. More importantly, they can also be used in creating new kernels or approximating the existing kernels through the Taylor series expansion [83]. Mathematically, it defines a covariance function between any two random vectors $x, z \, \epsilon \, \mathcal{R}^n$

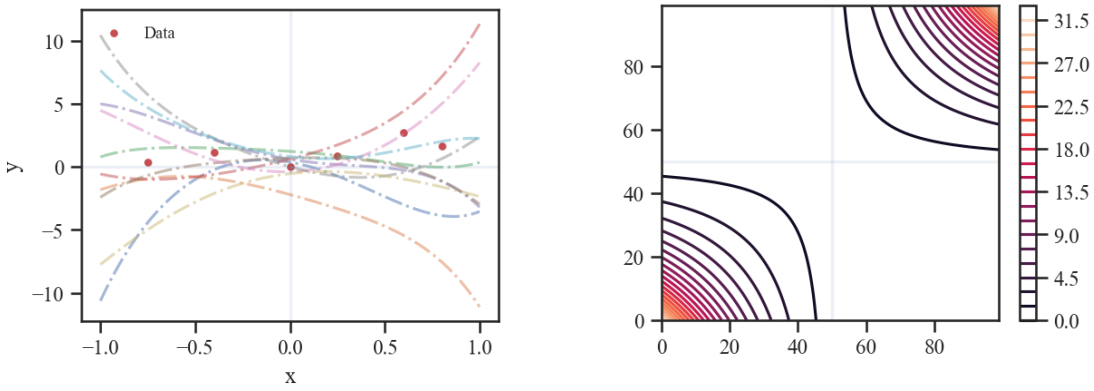
$$k(x, z) = (x^T z + c)^p \tag{2.11}$$

where c and p are the parameters defining the kernels. The parameter p determines the order of the polynomial function or the decision boundary in regression and classification tasks respectively. As such, it determines the degree of model complexity. The covariance map of a 5^{th} order polynomial kernel and few samples from the prior distribution of GP using it as a covariance function is shown in Figure 2.8. There is more freedom and flexibility when using the polynomial kernel. However, it is also the easiest kernel to overfit the given data during training.

2.4.4 Squared exponential kernel

The squared exponential (SE) kernel is the most commonly used similarity function in kernel-based learning models. It is a stationary kernel that defines a covariance function for any two vector x,$z \, \epsilon \, \mathcal{R}^n$

$$k(x, z) = exp\Big(- \frac{\|x - z\|^2}{2l^2} \Big) \tag{2.12}$$



(a) Polynomial kernel GP prior

(b) Polynomial kernel covariance map

Figure 2.8: GP prior distribution with 5^{th} order polynomial kernel and its covariance map

where $l > 0$ is the length parameter that controls the smoothness and complexity of the model or alternatively the spread of covariance. The similarity index is dictated by an exponentially decaying relative euclidean distance (i.e $\|x - z\|^2$) between data points regardless of their absolute position. Hence, smaller values of l will give high emphasis and assign a high value of similarity to points that are closer to a given point. At the same time disregard the effect of points that are farther away as if they are uncorrelated. As a result, the function returned will exhibit large variances or wiggleness with a smaller bias resembling the behaviour of a high-degree polynomial fitting. In contrary, a large value of l will have a sort of smoothing effect with a long range correlation to points near and far. As such, it assigns a similarity value to all points commensurate with their relative distance from the point under focus in a fashion similar to a low-order polynomial fitting would. In general, the parameterization of the kernel not only determines the complexity of the resulting model but also the contribution of each training point for the estimated value at the forecast point.

As a mapping function, they can project any two vectors x, $z \; \epsilon \; \mathcal{R}^n$ into an infinite dimensional space $\mathcal{R}^\infty$. For instance, assuming we have such a mapping function $\phi(\cdot)$

$$k(x,z) = \phi(x)^{'}\phi(z) = exp\Big(-\frac{\|x-z\|^2}{2l^2}\Big)$$
$$k(x,z) = exp\Big(-\frac{(x-z)^T \cdot (x-z)}{2l^2}\Big)$$

Expanding the exponential terms for $l = 1$

$$k(x,z) = exp\Big(-\frac{(x^Tx - 2x^Tz + z^Tz)}{2}\Big)$$
$$k(x,z) = exp\Big(-\frac{\|x\|^2}{2} - \frac{\|z\|^2}{2}\Big)exp\Big(2\frac{x^Tz}{2}\Big)$$

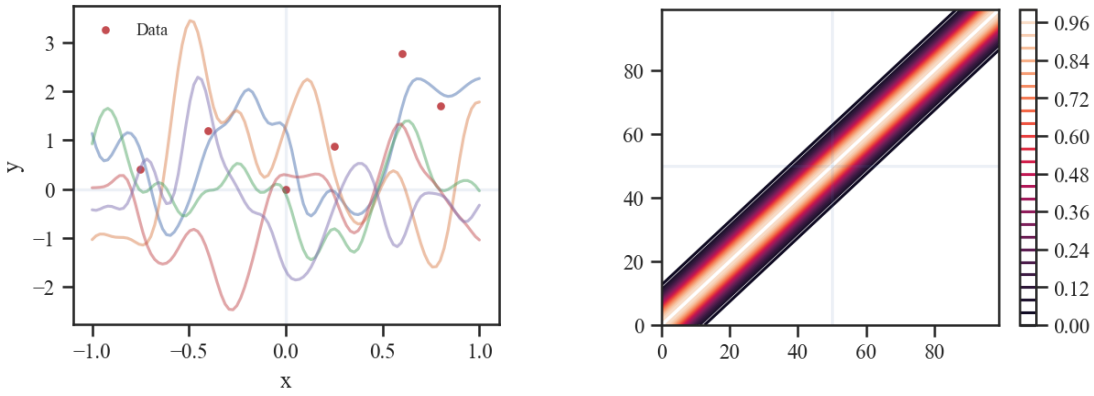
The first exponential term in the above equation is constant. Hence, assuming $\gamma = exp\Big(-\frac{\|x\|^2}{2} - \frac{\|z\|^2}{2}\Big)$

$$k(x,z) = \gamma exp\Big(x^Tz\Big)$$

Applying Taylor series expansion on $exp\Big(x^Tz\Big)$, it can be expanded into an infinite sum of polynomial kernels

$$\begin{aligned} exp\Big(x^Tz\Big) &= 1 + \beta_1(x^Tz) + \cdots + \beta_{n-1}(x^Tz)^{n-1} + \beta_n(x^Tz)^n + \cdots \\ k(x,z) &= \gamma exp\Big(x^Tz\Big) \Rightarrow \gamma \sum_{n=0}^{\infty} K_{poly}(x,z) \end{aligned} \tag{2.13}$$

The implication of equation (2.13) is that the squared exponential kernel is infinitely differentiable and very smooth. Consequently, in problems where function smoothness is a priority and the analysis of the derivative information is necessary, a gaussian process prior with a squared exponential kernel assumption can provide the desired qualities with an added benefits of derivative information of any order. The covariance map of SE kernel



(a) SE kernel GP prior

(b) SE kernel covariance map

Figure 2.9: GP prior distribution with SE kernel and its covariance map

and samples drawn from a prior distribution of GP with SE kernel is shown in Figure 2.9b and Figure 2.9a respectively. Figure 2.9a shows the smoothness of the resulting functions. However, its ability to introduce a smoothing effect presents a challenge when modeling a discontinuous function or modeling data with big gap in between data points. This is also reflected during prediction, when there is a large gap between the observed data and forecasted points, it produces a very smoothed function which might not reflect the behaviour of the underlying data generating process. In such moments a Matern class kernels that encapsulate the flexibility of SE kernels with an added benefits of modeling discontinuity can give a better performing models.

2.4.5 Matern kernels

These classes of kernels exhibit the properties of polynomial, exponential and to the extreme squared exponential kernel behaviours. Such diversity has given these kernels to excel in areas where the SE kernel couldn't. As a result, they are considered as a generalization to the SE kernel and by extension to the exponential kernel as well. They produce a less-smoother functions compared to the squared exponential kernel. However, if there is a possibility of missing data or discontinuous, a prior gaussian assumption with matern class kernels as a covariance function provides a better fitting models. The matern class kernel is a stationary kernel that defines a covariance function on any two vectors $x, z \, \epsilon \, \mathcal{R}^n$ which is given by

$$k(x,z) = \frac{1}{\Gamma(v)2^{v-1}}\Big(\frac{\sqrt{2v}}{l}\|x-z\|\Big)^{v} K_v\Big(\frac{\sqrt{2v}}{l}\|x-z\|\Big) \tag{2.14}$$

where, $\Gamma(v)$ is a gamma function, K_v is the modified Bessel function of the second kind and $\|x-z\|$ the euclidean distance between the two vectors. The kernel is parameterized by the length parameter $l > 0$ and $v > 0$. The purpose of parameter l is similar to the length parameter in squared exponential kernel. It control the range of influence a given data point has on points near and far as dictated by the decaying exponential euclidean distance between them. Parameter v also called the smoothing parameter, determines

the level of smoothness and flexibility of the kernel in modeling abrupt changes, discontinuities, gaps and variations in the data. As $v \to \infty$, the kernel resembles a squared exponential kernel resulting in a smoother functions with continuous differentiability. A lower value of v, results in a rougher function with discontinuous derivatives which helps in modeling data variations better than other alternative kernels. In practice values of v $\epsilon \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$ are frequently used and its approximation to the exponential and polynomial-exponential kernel is given in equation (2.15) respectively. The differentiability of the resulting function is dependent on the value of v. A Gaussian process with a matern covariance function is at most $v-1$ times differentiable. As such, for $v = \frac{1}{2}$ zero (discontinuous), for $v = \frac{3}{2}$ once and for $v = \frac{5}{2}$ is twice differentiable. Correspondingly, their reduced kernel equations become, for $v = \frac{1}{2}$

$$k^{\frac{1}{2}}(x, z) = exp\Big(-\frac{1}{l}\|x - z\|\Big)$$

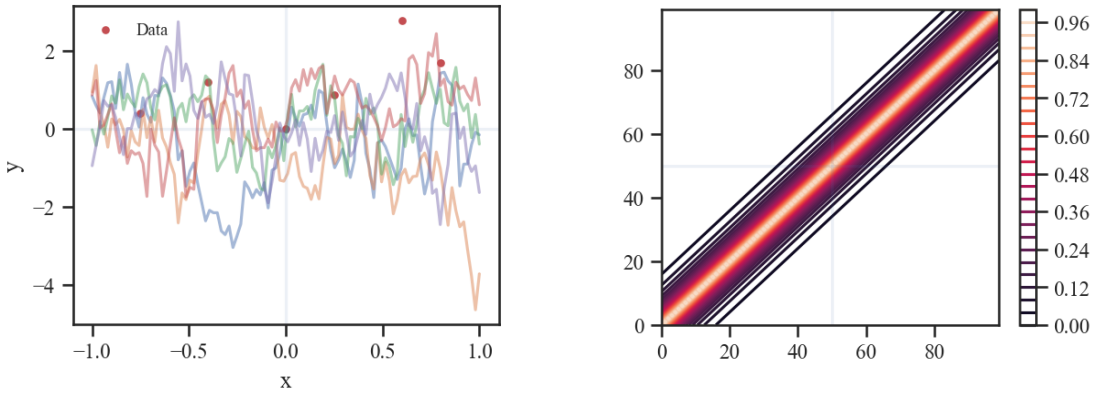
for $v = \frac{3}{2}$

$$k^{\frac{3}{2}}(x, z) = \Big(1 + \frac{\sqrt{3}}{l}\|x - z\|\Big)exp\Big(-\frac{\sqrt{3}}{l}\|x - z\|\Big) \tag{2.15}$$

for $v = \frac{5}{2}$

$$k^{\frac{5}{2}}(x, z) = \Big(1 + \frac{\sqrt{5}}{l}\|x - z\| + \frac{5}{3l}\|x - z\|^2\Big)exp\Big(-\frac{\sqrt{5}}{l}\|x - z\|\Big)$$

Optimization algorithms doesn't allow an abrupt change in model type or complexity during the optimization steps. Consequently, although the kernel has two parameters, only one of them (i.e the length parameter l) is trainable. As the parameter v determines the type of the model, it is a design parameter. Hence, it is pre-selected as a model specific parameter before training. Figure 2.10b and 2.10a show the covariance map and sampled functions from a GP prior with a matern kernel as a covariance function. The smoothing parameter was selected to be $v = 0.5$. Compared to the prior distribution of GP with SE kernel in Figure 2.9a, Figure 2.10a shows GP with a matern kernel produces a more rougher and sharp functions fit for modeling jumps. For $v = \frac{3}{2}, \frac{5}{2}, \cdots, \infty$ the resulting functions of the GP prior with matern kernel will resembles those shown in Figure 2.9a.



(a) Matern kernel GP prior

(b) Matern kernel covariance map

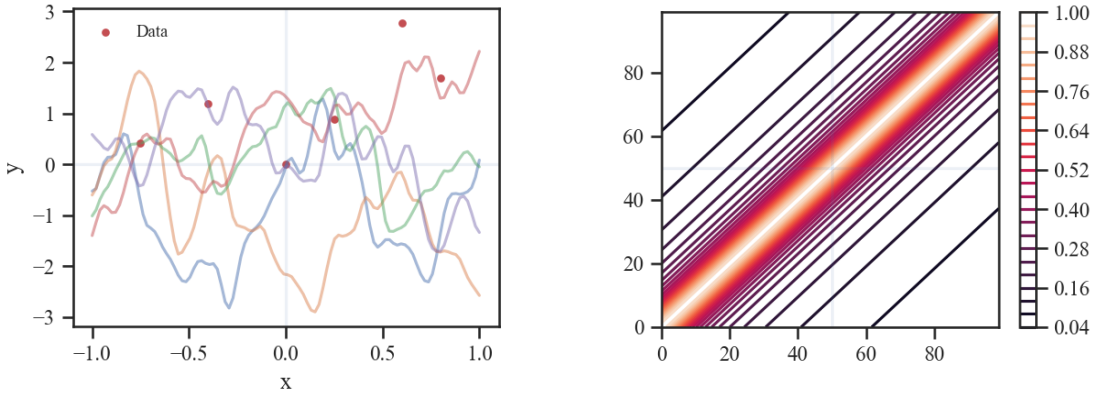
Figure 2.10: GP prior distribution with matern kernel *nu=0.5* and its covariance map

2.4.6 Rationale quadratic kernel

In the squared exponential kernel, the length parameter l control the range of influences a given point can have. Once fixed or optimized the model will either exhibit local dependence with high variance or a global dependence with low variance and high smoothing factor. The rational quadratic is a stationary kernel that encapsulate the benefits of a collection of SE kernels with varied length scales. It is represented as an infinite sum of SE kernels with differing length scales. As such, for any two vectors $x, z \, \epsilon \, \mathcal{R}^n$, it defines a covariance function given by

$$\begin{aligned} k(x,z) &= \int_0^\infty \Gamma(l) exp\Big(- \frac{\|x-z\|^2}{2l^2}\Big) \\ &\approx \Big(1 + \frac{1}{2\alpha l^2}\|x-z\|^2\Big)^{-\alpha} \end{aligned} \tag{2.16}$$

where, $\Gamma(l)$ is a gamma function, $\alpha > 0$ determines the weighting value applied on the



(a) RQ kernel GP prior

(b) RQ kernel covariance map

Figure 2.11: GP prior distribution with RQ kernel $\alpha = 0.5$,$l = 0.1$ and its covariance

different length parameter ($l > 0$) and $\|x - z\|$ represent the euclidean distance. Both the α and l parameters are trainable and optimised during training. A Gaussian process with a rational quadratic kernel as a covariance function produces a rougher priors for small values of α which is ideal for modeling abrupt data variations. As the values of $\alpha \to \infty$ the model will produce smoother functions with low variance just like the SE kernels would. The RQ kernel covariance map and sample functions from a GP that has a RQ kernel with $l = 0.1$ and $\alpha = 0.5$ as a covariance function are shown in Figure 2.11b and Figure 2.11a respectively. The depth of interaction between data points can be seen by comparing the covariance map of the SE kernel in Figure 2.9b and RQ kernel in Figure 2.11b. The consideration of varied mixture of length scales in rational quadratic has enabled points to interact with their nearby points as well as those farther away. However, the RQ kernel produces less smoother functions as shown in Figure 2.11a compared to the SE kernel in Figure 2.9a.

In general, the squared exponential, matern and the rationale quadratic kernels are great at interpolating and filling missing values regardless of the nature of the data. However, their extrapolation ability depends on the length parameter l and structure of the data. The estimated forecast of a Gaussian process that has any of those kernel as a covariance function will revert back to the mean value during extrapolation for a forecast horizon greater than the length parameter. Especially, for the data that exhibit periodic patterns, those kernels don't posses the framework to capture and set accurate similarity measures for the observed oscillatory behaviour [79].

2.4.7 Periodic kernels

Periodic kernels enables data points to interact intermittently. As a result, they provide an accurate similarity measure to points near and far but can affect one another recurrently. Consequently, the prediction of a Gaussian process with periodic kernel at a given point is determined by the contribution of all points that influences the given point intermittently.

Periodic kernel can be constructed from any stationary kernel by transforming the vectors $x, z \, \epsilon \, \mathcal{R}^n$ through a wrapper function $u(\cdot) = [cos(\cdot), sin(\cdot)]$ and then feeding the transformed vectors as an input to the stationary kernel. For instance, the exponentiated-sine squared periodic kernel uses the SE kernel as a base kernel. First, it transforms the given vectors x, z into $u(x) = [cos(x), sin(x)]$ and $u(z) = [cos(z), sin(z)]$ using the wrapper function respectively. Then, it substitutes the euclidean distance $\|x - z\|^2$ of the SE kernel with $\|u(x) - u(z)\|^2$.

Assuming there is such function $u(\cdot) : x, z \mapsto [cos(\cdot), sin(\cdot)]$, the euclidean distance between the vectors is given as

$$\begin{aligned}
\|u(x) - u(z)\|^2 &= \Big(u(x) - u(z)\Big)^T \Big(u(x) - u(z)\Big) \\
\|u(x) - u(z)\|^2 &= \Big(cos(x) - cos(z)\Big)^2 + \Big(sin(x) - sin(z)\Big)^2 \\
\|u(x) - u(z)\|^2 &= 2\Big(1 - cos(x - z)\Big)
\end{aligned}$$

Applying the half angle formula on $2\Big(1 - cos(x - z)\Big)$

$$\|u(x) - u(z)\|^2 = 2\Big(2sin^2(\frac{x - z}{2})\Big)$$

Considering periodic recurrence $2\pi f$ or $\frac{2\pi}{p}$, the above equation can rewritten as

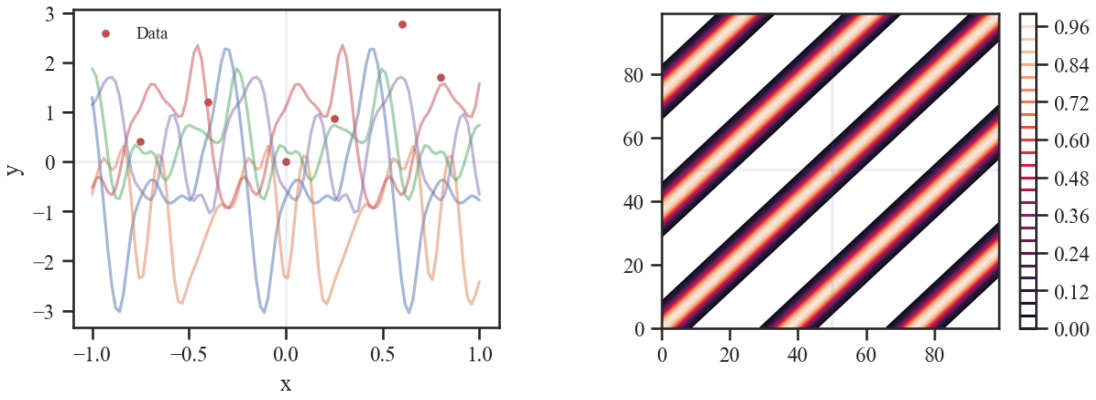
$$\|u(x) - u(z)\|^2 = 2\Big(2sin^2(\frac{\pi(x - z)}{p})\Big)$$

Now given the SE kernel $k(x, z) = exp\Big(-\frac{\|x-z\|^2}{2l^2}\Big)$ and substituting $\|u(x) - u(z)\|^2$ in place of $\|x - z\|^2$

$$\begin{aligned}
k(x, z) &= exp\Big(-\frac{\|u(x) - u(z)\|^2}{2l^2}\Big) \\
k(x, z) &= exp\Big(-\frac{2\Big(2sin^2(\frac{\pi(x-z)}{p})\Big)}{2l^2}\Big) \Rightarrow exp\Big(-\frac{\Big(2sin^2(\frac{\pi(x-z)}{p})\Big)}{l^2}\Big)
\end{aligned} \tag{2.17}$$

where l and p are the length parameter and the fundamental period that characterizes the cyclic event respectively. The periodic kernel covariance map and a few sampled functions from a GP model with a periodic kernel are shown in Figure 2.12b and 2.12a. The length parameter determines the level of smoothness of the resulting functions and the extent of interaction as shown in prior distribution and the covariance map of Figure 2.12a and Figure 2.12b respectively. Small values of l produces narrower strip and large values produce broad strip in the covariance map to signify the level of influence. The parameter p is the fundamental period. Its value determine the number of strip in the covariance map. As such, for small frequency values, we are bound to see a lot of strips on the covariance map.

Note:, We have been referring the variables that describe the dynamics of a kernel as parameters. Since they directly impact the response of the kernel, the parameter reference is a fitting description. However, it should be noted that when discussing kernel-based learning models, the same parameters will be addressed as hyper-parameters due to their indirect influence through the kernel. Despite the differing nomenclature used, both references point to the same variables.



(a) Periodic kernel GP prior

(b) Periodic kernel covariance map

Figure 2.12: GP prior with periodic kernel $p = 0.75$, $l = 0.5$ and its covariance map

2.4.8 Compositional kernels

The tendency to base forecast decisions based on the wisdom of crowed sourcing is the advantage that the Bayesian and deep learning models share in common. These models capitalize on the information gathered from numerous sources to make better decision. They weigh the contributions of different functions and the appropriateness of their solution to a given problem. The more features the model can distinguish, the better will be its generalization and forecast accuracy. Hence, learning and inference at its core depends on the model's ability to understand and discern the structure of the data. For instance, a neural network that is designed to differentiate linear features, can be made to learn non-linear mappings with an additional hidden layer. The extra layer improves the network's flexibility and competence to analyze new features. Consequently, its output will be the cumulative result of all details that the network was able to detect. The exploration of complex features in kernel-based learning models also include the utilization of multiple kernels. Individually, the kernel functions discussed in section 2.4 may not be adequate in representing the full spectrum of patterns observed in a given data. However, pattern mapping by combining a number of kernel together presents an opportunity to utilize the combined predictive performance of all. For instance, SE kernel mostly capture trends and local similarity where as the periodic kernel looks for global interdependence corresponding to the fundamental frequency. The additive or multiplicative combination of these kernels undeniably would give improved performance than either of them used alone. As a result, various kernel learning schemes such as analytical [79], data-driven [84], numerical approximation [60, 83, 85] have been put forth as an alternative approach to kernel design or enhancing its fitness.

Some of the major challenges associated with kernel learning is the question of how best to select it and the algorithm that should be used for optimising its parameter [86]. Practically, we mostly make personal decision on which kernel and algorithm to apply or think are more appropriate to the task at hand. The selection signify that we implicitly put a restriction on the relevant features the model should focus on learning. In doing so, the assumption will inadvertently bias the outcome [87]. In fact this goes against the aim of machine learning. The selected kernels may or may capture the best features. One possible strategy in rectifying this limitation is proposing an algorithm which is capable of automatically selecting the relevant features through a combination of predefined collection of kernels [81, 86, 87].

Various reasons can be given for the motivation behind combining kernels. For instance, it can be the need to explore and understand the structure of the data, handle

heterogeneous information fusion or improve the predictive performance of the data generation model. Some other time, we might find the exiting kernels are perhaps computationally intensive. As such, we needed a new approach to approximate or potentially replace it by another kernel that is more efficient [83]. Regardless of the reasons or the method of combination, all kernels, old and new, should provide an acceptable measure of similarity in accordance with the task and the nature of the data. They should also adhere to the characteristic of a kernel given in proposition 2.4.1 and the principles of kernel combination stated in proposition 2.4.2. To that end, various multiple kernel learning (MKL) algorithms differing in principles and optimisation techniques (i.e rule based, heuristic, Bayesian, optimisation, boosting) have been implemented [87]. The rule based kernel combination follows the sum and product rule given in proposition 2.4.2 to create new kernels. According to it, for any two predefined kernel functions k_1 and k_2, K is a new kernel given by

$$\begin{aligned} \mathrm{K} &= \alpha_1 k_1 + \alpha_2 k_2, \quad \textit{for any } \alpha_1, \alpha_2 > 0 \\ \mathrm{K} &= \alpha_1 k_1 * \alpha_2 k_2, \quad \textit{for any } \alpha_1, \alpha_2 > 0 \end{aligned} \tag{2.18}$$

The coefficients α_1 and α_2 are mostly substituted by a constant kernel. Such approaches prefer to optimise all parameters as part of the fitting process and no particular optimisation step is carried out separately for the coefficients. Although this scheme appears to be simple, the selection of suitable kernels for combination requires expert domain knowledge in regards to the type of data and the problem in general. Especially, for a time-series data that exhibit visible patterns, this methods offers an interpretable kernel structure with acceptable predictive performance. Another alternative that is proposed for kernel learning is to present kernel combination as a convex optimisation problem and leverage the objective function convexity in order estimate the coefficient's for the mixture [87]. Given base kernels $k_1, \cdots, k_m$ and parameter vector $\eta \, \epsilon \, \mathcal{R}^m$, a new kernel $\mathrm{K}(\eta)$ can be defined as

$$\begin{aligned} \mathrm{K}(\eta) &= \sum_{i=1}^{m} \eta_i k_i \\ \eta &= \arg\min_{\eta \, \epsilon \, \mathcal{R}^m} \mathrm{J}(\eta) \end{aligned} \tag{2.19}$$

The parameterization of the cost function $\mathrm{J}(\eta)$ varies depending on the specialization. For instance, in regularised linear regression, the objective function is given as $J(\eta) = \frac{\lambda}{2} y^T (K(\eta) + n\lambda I)^{-1} y$, where n and λ are number of points and regularization factor respectively. Despite the mathematical form, the parameter η's values are evaluated by minimizing the given cost function $\mathrm{J}(\eta)$. The Bayesian methods takes a robust and probabilistic approach to optimal weight assignment by considering the combination parameters as random vectors and defining a prior distribution on $\eta \sim Dir(\eta; \alpha)$ where $\alpha \sim \Gamma(k, \beta)$. The resulting posterior will be used as the coefficient for the kernel combination in equation (2.19). There are also performance based weight assignments where the predictive quality of one kernel is used as a based to to scale the weights for the whole combination [86, 87]. Alternatively, the boosting or ensemble based approaches follow a recursive addition of base kernels to the combination until a certain performance index is met where the value of the coefficients are optimised during turning along with the kernel parameters.

The multiple kernel learning frameworks undoubtedly give machine learning models the capability to process complex features, analyze and fuse information from various sources. However, such degree of flexibility comes at a cost of negatively impacting the computational performance of the underlying model. For example, combining kernels compounds the number of parameters to be optimised. As the number of kernels increase, the computational complexity of the model will also increase. As a result, the time

and resource requirement for parameter optimisation and model fitting will be higher. Additionally, a higher number of kernels in model training can lead to a complex model that is highly vulnerable to overfitting, resulting in a model that performs well during training and worse on validation. Furthermore, algorithmic based kernel combination puts model interpretability at risk. The reason being, the algorithms automatically combines kernels to meet some predefined performance metrics among which model interpretability is not one of them. To that end, they construct complex relationships between inputs and outputs features in order to achieve it. Unfortunately, due to the inherent nature of the algorithm, the interpretability of the final model is not guaranteed. Especially, in areas where high model transparency is a requirement, such procedure presents a challenge resulting in the undesirability of the trained model.

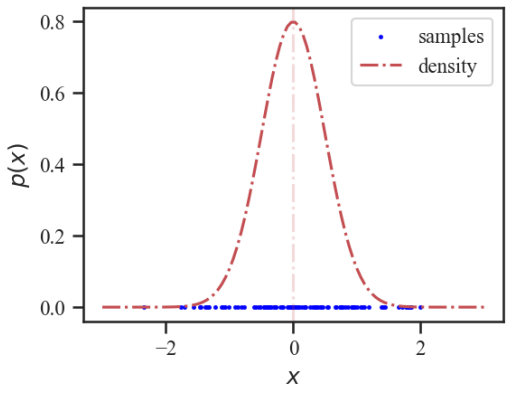
2.5 Gaussian process probabilistic models

Probabilistic models tends to predicts the trajectory of the mean and its variance. This is due to the high susceptibility of individual behaviour to variation. As such, modeling every singular randomness is very difficult. However, the behaviour of the average population although it varies in its own pace, stays minimal compared to the samples variations. Hence, analysing and predicting group behaviour is much easier. In the collective assessments of the mass, the effect of individual variability is minimal [88].

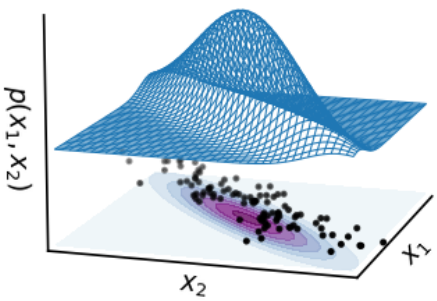
In this subsection, we present the mathematical concept behind the multivariate Gaussian process model. The Gaussian distribution has been widely used in machine learning and system modeling. The reason behind its popularity can be attributed to its finite-dimensional sufficient statistics. Meaning that the number of parameters needed to describe the observed data and characterize the behavior of the corresponding distribution do not increase with the size of the data. Hence, the distribution requires a fixed set of parameters that can sufficiently define it. Figure 2.13a shows a univariate Gaussian distribution for a random variable x with mean μ and variance σ^2. Its probability density $p(x|\mu, \sigma^2)$ is given by

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp\Big(- \frac{(x-\mu)^2}{2\sigma^2}\Big) \tag{2.20}$$

The parameters μ and σ^2 contain all the accessible information that can be inferred from



(a) Univariate gaussian density plot



(b) Bivariate gaussian density plot

Figure 2.13: Gaussian univariate and bivariate distribution

the given data as well as the future expectations from the generation model. In addition to compressing the information content of a data, the mean-variance parameterization simplify the process of mapping one distribution into another. For instance, estimations in areas like variational inference, stochastic sampling and normalizing flows involve transforming distributions from one form to another in order to study variables from a different perspective or use the existing distribution as base to create another. Regardless of the form of the distribution, this transformation has been carried out either through the cumulative distribution function (CDF) or the change of variable techniques which requires a bit of computations. In case of Gaussian, however, due to the mean-variance parameterization complex distributions can easily be constructed by transforming the mean μ and variance σ^2 parameters. This approach doesn't require applying the CDF or rules of change of variables. This has been the reason why Gaussian linear transformation rules are widely preferred in posterior derivation than the approach suggested in the Bayesian framework. As a result, a Gaussian is defined as a distribution that is fully specified by its mean and variance parameters [82].

In probabilistic linear regression, the univariate Gaussian has been used to analyse the cause-and-effect relationship between variables and accommodate different features. However, those features and their interactions are modeled only as part of the mean parameter. Being univariate, the distribution doesn't offer the framework to model the relative correlations between multiple random variables. In such moments, the multivariate Gaussian distribution can provide a valuable insight into how the correlation between the random variables impact the predictive performance of a model.

The multivariate Gaussian distribution is an extension and a generalization to the univariate Gaussian in a higher dimension which is used to study the correlations among multiple random variables. Consequently, a vector of n random variables is said to be n-variate Gaussian distributed, if every linear combination of its n components is also normally distributed. That is, given a vector of random variables $\mathrm{X} = \{x_1, x_2, \cdot, \cdot, x_n\} \; \epsilon \; \mathcal{R}^n$ is multivariate Gaussian if, for every $\beta = \{\beta_1, \beta_2, \cdot, \cdot, \beta_n\} \; \epsilon \; \mathcal{R}^n$, their linear combination $\phi(\mathrm{X}, \beta) = \sum_{i=1}^{n} \beta_i x_i$ is Gaussian. If so, their joint distribution can be described by

$$\begin{aligned} p(x_1, \cdot\cdot, x_n | \mu, \Sigma) &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} exp\Big(-\frac{1}{2} (\mathrm{X} - \mu)^T \Sigma^{-1} (\mathrm{X} - \mu) \Big) \\ &\sim \mathcal{N}(\mu, \Sigma) \end{aligned} \tag{2.21}$$

where $\mu \; \epsilon \; \mathcal{R}^n$ is now a mean vector and $\Sigma \; \epsilon \; \mathcal{R}^{nxn}$ is a symmetric semi-definite covariance matrix with $|\Sigma|$ as its determinant. Visualising a multidimensional distribution for $n > 2$ is difficult. However, for two random vectors x_1 and x_2 a bivariate joint distribution can be defined as

$$\begin{aligned} p(x_1, x_2 | \mu, \Sigma) &= \frac{1}{(2\pi) |\Sigma|^{1/2}} exp \left(-\tfrac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right) \\ \text{where,} \quad \Sigma^{-1} &= \begin{bmatrix} \Sigma_{x_1 x_1} & \Sigma_{x_1 x_2} \\ \Sigma_{x_1 x_2} & \Sigma_{x_2 x_2} \end{bmatrix}^{-1} \end{aligned} \tag{2.22}$$

where $\mu = [\mu_1, \mu_2]^T$ is the mean vector, $\Sigma_{x_1 x_1}$ & $\Sigma_{x_2 x_2}$ describe the marginal variance of x_1 & x_2 and $\Sigma_{x_1 x_2}$ models the covariance between them. Figure 2.13b shows the bell-shaped joint density, covariance contour plot and few sampled data points for vectors distributed according to $p(x_1, x_2) \sim \mathcal{N}(0, [[1, 0.9], [0.9, 1]])$.

2.5.1 Multivariate distribution in probabilistic regression

The general objective in regression is to model the relationship between the inputs features x_n and the target variable y_n.

$$y_n = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{2.23}$$

The multivariate distribution provides a framework that simplifies the analysis in a way that captures the mutual interaction between the given predictors through probabilistic distributions. For instance, the Bayesian linear regression takes a probabilistic approach on model outputs and parameters by assuming both quantities as random vectors that follow a multivariate normal distribution. For instance, given a time series data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ and considering D-input features, the target vector Y is assumed to follows a Gaussian distribution

$$\begin{gathered} \mathrm{Y} = f(\mathrm{X}) + \epsilon \quad where \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I_d) \quad , \quad f(\mathrm{X}) = \mathrm{X}\beta \\ X = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & ... & x_1^d \\ 1 & x_2 & x_2^2 & x_2^3 & ... & x_2^d \\ & & & ... & & \\ & & & ... & & \\ & & & ... & & \\ 1 & x_n & x_n^2 & x_n^3 & ... & x_n^d \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_d \end{bmatrix} \end{gathered} \tag{2.24}$$

where, the vector X represents the design matrix for the input features, β the parameters of the model and ϵ the perceived aleatoric uncertainty. Applying the linear Gaussian transformation rule, the likelihood function can also be written as a distribution

$$\begin{aligned} \mathrm{Y} &= \mathrm{X}\beta + \epsilon \\ Y &\sim \mathcal{N}(X\beta, \sigma^2 I_d) \end{aligned} \tag{2.25}$$

For N observations, the joint distribution becomes an N-dimensional multivariate distribution with mean $X\beta$ and variance $\sigma^2 I_d$ whose density is given by

$$p(y_1, y_2, \cdot, \cdot, y_n | X, \beta, \sigma) = \frac{1}{(2\pi)^{N/2} |\sigma^2 I_d|^{1/2}} exp\Big(-\frac{1}{2}(y - X\beta)^T \sigma^{-2} I_d (y - X\beta)\Big) \tag{2.26}$$

Since the likelihood is a Gaussian, for mathematical convenience a Gaussian prior is assumed for the parameter distribution. Hence, the Gaussian prior for β is also given by $p(\beta) \sim \mathcal{N}(\mu_o, \Sigma_o)$. For a D-input features, this also defines a D-dimensional joint parameter distribution

$$p(\beta_1, \beta_2, \cdot, \cdot, \beta_d | \mu_o, \Sigma_o) = \frac{1}{(2\pi)^{D/2} |\Sigma_o|^{1/2}} exp\Big(-\frac{1}{2}(\beta - \mu_o)^T \Sigma_o^{-1} (\beta - \mu_o)\Big) \tag{2.27}$$

Equation (2.27) makes it possible to model the epistemic uncertainty through the parameter covariance matrix Σ. Since both the likelihood and prior follow a normal distribution, the parameter posterior will also be gaussian $p(\beta | Y, X, \mu_o, \Sigma_o) \sim \mathcal{N}(\mu_p, \Sigma_p)$ and it can be estimated by applying Bayesian inference on equation (2.26) & (2.27),

$$p(\beta | Y, X, \mu_o, \Sigma_o) = \frac{p(Y | X, \beta, \sigma) p(\beta | \mu_o, \Sigma_o)}{\int p(Y | X, \beta, \sigma) p(\beta | \mu_o, \Sigma_o)} \tag{2.28}$$

The integral in equation (2.28) complicates the derivation. The easiest alternative is to

apply moment matching on LHS and RHS of equation (2.28)

$$
\begin{aligned}
p(\beta|Y,X,\mu_p,\Sigma_p) &= \frac{p(Y|X,\beta,\sigma)p(\beta|\mu_o,\Sigma_o)}{\int p(Y|X,\beta,\sigma)p(\beta|\mu_o,\Sigma_o)} \\
p(\beta|Y,X,\mu_p,\Sigma_p) &\;\alpha\; p(Y|X,\beta,\sigma)p(\beta|\mu_o,\Sigma_o) \\
\frac{1}{(2\pi)^{D/2}|\Sigma_p|^{1/2}} exp\Big(-\frac{1}{2}(\beta-\mu_p)^T\Sigma_p^{-1}(\beta-\mu_p)\Big) &\;\alpha\; \frac{1}{(2\pi)^{N/2}|\sigma^2 I_d|^{1/2}} * \\
exp\Big(-\frac{1}{2}(y-X\beta)^T(\sigma^2 I_d)^{-1}(y-X\beta)\Big)\frac{1}{(2\pi)^{D/2}|\Sigma_o|^{1/2}} exp\Big(-\frac{1}{2}(\beta-\mu_o)^T\Sigma_o^{-1}(\beta-\mu_o)\Big)
\end{aligned}
\tag{2.29}
$$

collecting terms only related to β and matching the RHS with the LHS of equation (2.29), the posterior parameter distribution can be given as

$$
\begin{aligned}
\Sigma_p &= \Big(\Sigma_o^{-1} + \frac{1}{\sigma^2}X^TX\Big)^{-1} \\
\mu_p &= \Big(\Sigma_o^{-1}\mu_o + \frac{1}{\sigma^2}X^Tt\Big)\Sigma_p
\end{aligned}
\tag{2.30}
$$

where the μ_p and Σ_p represent the posterior mean and variance distribution for the parameter β. Hence, using equation (2.24) and (2.30), the target posterior distribution can be computed

$$
\begin{aligned}
f(Y) &= X\beta + \epsilon, \quad where \\
\epsilon &\sim \mathcal{N}(0, \sigma^2 I_d) \\
\beta &\sim \mathcal{N}(\mu_p, \Sigma_p) \\
&Applying\ the\ linear\ transformtion\ rule \\
f(Y) &\sim \mathcal{N}(X\mu_p, X^T\Sigma_p X + \sigma^2 I_d)
\end{aligned}
\tag{2.31}
$$

Equation (2.31) provides the model's output distribution with mean $X\mu_p$ and variance $X^T\Sigma_p X + \sigma^2 I_d$. In addition to providing a forecast distribution, the estimation methods follows a principled approach in addressing the aleatoric uncertainty through σ^2 and the epistemic uncertainty with Σ_p as shown in equation (2.31). Bayesian linear regression through the multivariate distribution context allowed it to encompass the impact the inputs correlation has on the mean trajectory and estimated variance. Something which was difficult to do in the univariate case. However, due to its parametric nature, the data generation capability or the accuracy of the model in representing the observed data is still constrained by the number of input features or the number of parameters considered. Furthermore, since the selection of the features (i.e the complexity of the model) is a design parameter, the suitability of the model to the given problem remains subjective. On the contrary, the Gaussian process is the non-parametric version of Bayesian inferential learning that makes no prior such assumption on the number of parameters required to fully define a model. In a multivariate distribution context, it relies on the prior and posterior distribution of functions in contrary to the parametric distribution followed in the case of parametric regression models. As such, the Gaussian process is a multivariate distribution over functions. Consequently, this data-driven approach to model building gives the Gaussian process an advantage in delivering an appropriate fit functions that is relevant to the problem at hand compared to other regression methods.

2.5.2 Gaussian process regression

Definition 2.5.1. *Formally Gaussian process model is defined as a stochastic process that maps every input x_i to a random function $f(x_i)$ where the joint distribution $p(f(x_1), \cdot, f(x_n))$*

of a finite collection of these random functions $f(x) = \{f(x_1), f(x_2), \cdot\cdot, f(x_n)\}$ *is a multivariate Gaussian.*

As a distribution over functions, it is fully specified by a mean and covariance functions. Consequently, it follows a multivariate distribution parameterized with a mean $\mu(x)$ and covariance matrix Σ,

$$p(f(x_1), f(x_2), \cdot\cdot, f(x_n)) \sim \mathcal{N}(\mu(x), \Sigma) \quad \text{, where}$$
$$\mu(x) = \begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \cdot \\ \cdot \\ \cdot \\ \mu(x_n) \end{bmatrix} \text{, } \Sigma(x,x) = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdot & \cdot & \cdot & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \cdot & \cdot & \cdot & k(x_2,x_n) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ k(x_n,x_1) & k(x_n,x_2) & \cdot & \cdot & \cdot & k(x_n,x_n) \end{bmatrix}, \tag{2.32}$$

where the covariance matrix Σ is a symmetric positive semi-definite matrix that is also referenced as the kernel matrix. It's elements are evaluated based on the specific kernel function $k(x,x)$ employed. Data normalization and mean-centring can also give an alternative specification where the mean function is assumed to be zero. In fact, the Gaussian process with zero mean $P(f_1, f_2, \cdot\cdot, f_n) \sim \mathcal{N}(0, \Sigma)$ is the most widely used form of representation.

2.5.3 Prior distribution

In Gaussian process model, the Bayesian framework treats functions as random variables. The prior distribution quantify the belief and understanding on the distribution of those functions before observing any data. How inclusive and informative the prior, is entirely dependant on the past observed history and level of expertise. The information about the structure of the data, the patterns it is exhibiting and the possible functions that are expected to sufficiently represent it, are primarily encoded through a selection of the kernels. In return, these choices makes the prior definition more or less subjective. And yet, no more subjective than the various assumption we make about the nature of the anticipated noise, the number of parameters or their distribution, when building other predictive models.

As a non-parametric model the Gaussian process maps every input x_i to a random variable output y_i. Meaning that for infinite number of inputs X_∞, we will have infinite number random variables Y_∞ which makes the underlying model infinite dimensional. Something which exceeds the bound of realization. However, according to Definition 2.5.1, a collection of those random variables are jointly Gaussian. As such, bounding the collection of the observed values within a finite set, it is possible to create a distribution that is computable. Furthermore, by dividing the collection into random variables at the observed location f and random variables at any other location f_*, the prior distribution of the Gaussian process can be established. Hence, for a time series data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ the Gaussian process prior distribution over functions can be formally defined as a joint distribution $p(f, f_*)$ over the random variables f at observed point x and the random variable f_* at all other points x_* as

$$p\left(f, f_*\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{ff} & \Sigma_{ff_*} \\ \Sigma_{f_*f}^T & \Sigma_{f_*f_*} \end{bmatrix}\right) \tag{2.33}$$

where, μ is assumed to be zero. Whereas, $\Sigma_{ff}, \Sigma_{f_*f_*}$ and $\Sigma ff_* = \Sigma_{f_*f}$ models the marginal and cross-variance between the points at observed x and test points x_* respectively. Assuming there are N observations and P forecast points, Σ_{ff}, $\Sigma_{f_*f_*}$ and Σ_{ff_*} are an N x N, P x P and N x P sized matrices respectively.

Applying the multivariate marginalization rule on equation (2.33), the marginal distribution at the observed points $p(f)$ and at test points $p(f_*)$ is given as

$$\begin{aligned} p(f) &\sim \mathcal{N}(0, \Sigma_{ff}) \\ p(f_*) &\sim \mathcal{N}(0, \Sigma_{f_*f_*}) \end{aligned} \tag{2.34}$$

Marginal probabilities are used to compute the probabilities and expectations of a singular random variable. However, if the desired outcome is the probabilities of more than one random variable, then a joint probability distribution is used instead. Marginal distributions contain relevant information regarding a specific random variable under consideration. As such, they don't contain enough information to completely specify the joint distribution of one random variable with respect to another. Contrarily, the joint distribution contains additional information regarding the cause-effect relationship between random variables, an information which is not available in the case of marginal distribution. For instance, given the prior distribution a Gaussian process model in equation (2.33), the joint probability density between the random variables f and f_* is given as

$$p(f, f_*) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma|^{1/2}} exp\left(-\tfrac{1}{2}\begin{bmatrix} f \\ f_* \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} f \\ f_* \end{bmatrix}\right) \tag{2.35}$$

2.5.4 Posterior distribution

Bayesian learning evolves only after observing new information. This change in perspective is manifested either by changing the parameter distribution or the prior understanding of the problem at hand. Even then, Bayesian models are largely skeptical. For instance, if an N-time coin flip experiment resulted in N number of heads and 0 tails outcomes, the frequentist would believe that the probability of the next outcome being head is 1. However, being skeptical even in light of the observed data, the Bayesian model wouldn't assume the probability of the next outcome being head as 1. And yet, they are not close-minded either. If they are presented with an evidence-based reasoning, then the Bayesian model would be compelled to recreate a posterior-self. In the context of distribution, this posterior-self is referenced as the posterior distribution. The posterior contains valuable information regarding how likely some values are after observing new data and the degree of uncertainty associated with that assertion. Mathematically, its estimation is conditioned on the observed data, the likelihood and prior distribution. As such, before driving the posterior distribution of the latent random variables, lets take a look at the likelihood of the observed data under the Gaussian process.

In latent variable models the observed data $y_1, y_2, \cdot, \cdot, y_n$ are assumed to be a realization or samples drawn from another random variable Y that is distributed according to

$$\begin{aligned} \mathrm{Y} &= f + \epsilon \text{ , where, } \epsilon \sim \mathcal{N}(0, \sigma^2 I_d) \\ \mathrm{Y} &\sim \mathcal{N}(f, \sigma^2 I_d) \end{aligned} \tag{2.36}$$

In equation (2.34) the marginal distribution $p(f)$ of the random variable at observed point is given. Applying the Gaussian linear transformation and marginalizing f, equation (2.36) can be re-written as

$$\begin{aligned} \mathrm{Y} &\sim \mathcal{N}(f, \sigma^2 I_d) \\ \mathrm{Y} &\sim \mathcal{N}(0, \Sigma_{ff} + \sigma^2 I_d) \end{aligned} \tag{2.37}$$

Since f is marginalized, it is now called the latent variable, as its effect is felt but not observed. Equation (2.37) provides the distribution of the random variable Y at observed

points along with the expected uncertainty. Hence, it can be used to define the likelihood for the time series data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ as

$$p(y|f, x, \theta) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_{ff} + \sigma^2 I_d|^{1/2}} exp\left(-\tfrac{1}{2} y^T (\Sigma_{ff} + \sigma^2 I_d)^{-1} y\right) \tag{2.38}$$

where Σ_{ff} is the covariance matrix for the observed points, θ the kernel parameters, σ^2 is the assumed aleatoric uncertainty and N is the number of training samples. The likelihood function in equation (2.38) is mostly used for optimising the parameters θ defining the contents of the covariance matrix Σ_{ff} as part of the model fitting process.

2.5.4.1 Gaussian process conditionals

Multivariate distribution encapsulate the interaction of multiple random variables. Although being stochastic implies the possibility of acquiring any value at any given time, it doesn't imply that the values of the random variable will observable all the time. In such moments, the best approach is to condition the probable values the variable could take, based on the values of the other variables it correlated with. The multivariate conditional distribution offers an opportunity to guess the probable values a random variable could take given the observed values of other variables. Even if they don't provide accurate information regarding the values acquired by the variable, they could provide a probable value from the knowledge of other observed variables. However, such inference is only possible if there is an interdependence between the random variables. If the random variables are independent, no amount of data about the other variables will give us adequate information regarding the variable in question. In such scenarios, the random variables are said to be independent.

The Gaussian process prior distribution establish the required dependency between the latent variables at testing points f_* and at observed points f within its covariance matrix, so that the former can be predicated conditioned on the values of the later. To that end, from equation (2.34) and (2.37), we have the marginal distribution of the random variables f_* and the observed data Y respectively. These equations can be combined and used to define the joint distribution between the observed data Y and f_* as

$$p\left(\mathrm{Y}, f_*\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{ff} + \sigma^2 I_d & \Sigma_{ff_*} \\ \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}\right) \tag{2.39}$$

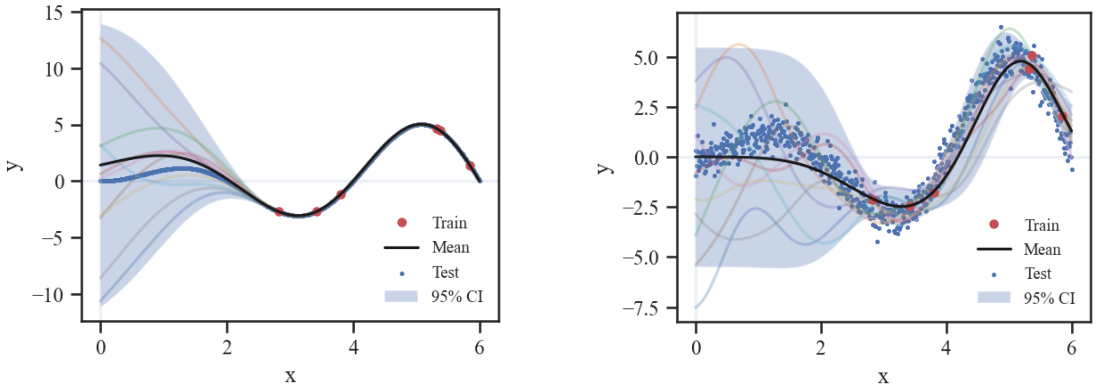
Applying the multivariate conditional rule [see Appendix. 5] on equation (2.39), the Gaussian conditional at test points or more formally the posterior predictive distribution of the model at forecast point is given by

$$\begin{aligned} p(f_*|f, x, y, \theta) &\sim \mathcal{N}(\mu_*, \Sigma_*)\ , where \\ \mu_* &= \Sigma_{f_*f}(\Sigma_{ff} + \sigma^2 I_d)^{-1} y \\ \Sigma_* &= \Sigma_{f_*f_*} - \Sigma_{f_*f}(\Sigma_{ff} + \sigma^2 I_d)^{-1}\Sigma_{ff_*} \end{aligned} \tag{2.40}$$

Note: the variance of the conditional distribution $p(f_*|f, x, y, \theta)$ in equation (2.40) is the Schur complement of the block $\Sigma_{f_*f_*}$ of the covariance matrix given in equation (2.33). Equation (2.40) shows the predicted variance Σ_* doesn't depend on values of the observed variable Y, but rather on the relative distance between observations [89]. As a result, the model confidence in its prediction is higher (i.e narrow confidence interval) for forecast points that are closer to the observed values as shown in Figure 2.14. On the other hand, the predicted mean $\mu_* = \Sigma_{f_*f}(\Sigma_{ff} + \sigma^2 I_d)^{-1} y$ resembles a weighted average combination of all y values (i.e $\mu_* = By$, where $B = \Sigma_{f_*f}(\Sigma_{ff} + \sigma^2 I_d)^{-1}$), where the weights depend on the prior distribution and the selected kernel function.

In regards to the uncertainty quantification, equation (2.40) considers both sources of uncertainty. Like in the Bayesian linear regression, the model tries to anticipate variability through the data variance σ^2 and modeling inadequacy or model variance with Σ_{ff} while evaluating a fitting function. Equation (2.33) - (2.40) show the significance of the covariance matrix in determining the suitability of the model to a given problem. The Gaussian process prior, likelihood and posterior distribution are all dependent on this matrix. By extension, the dimension of the matrix and the specific kernel functions used impact the computational efficiency and predictive accuracy of the underlying model.

For instance, assuming we have a data generation model given by $y = x * sin(\frac{\pi}{2}x)$ for an $x \epsilon [0, 6]$ and six observed points $(x, y) \epsilon \{(5.8, 1.9), (5.3, 4.6) , (2.8, -2.9), (5.3, 4.6), (3.4, -2.9), (3.8, -0.5)\}$. A SE kernel for covariance mapping and a noise variance of $\sigma^2 = 0.5^2$ were selected to create a noise-free and noisy Gaussian process models. The prior distribution of the models resembles a Gaussian process with an SE kernel given in Figure 2.9a. However, the posterior distribution depends on training points, the optimised SE hyperparameters and the absence or presence of a noise. Figure 2.14a and Figure 2.14b show the posterior conditional distribution for a noise-free and noisy GP model with an assumed data variance of $\sigma = 0.5^2$. The Figures clearly show in areas of adequate observed data, both the noisy and noise-free models predict with more confidence (i.e narrow prediction interval), with the exception that the noisy prediction takes into account data variability. As a result, the prediction interval width of the noisy GP is higher compared to the noise-free GP as shown in Figure 2.14. It should be noted that a prediction with an SE kernel will go back to the mean of the training data $\mu(x) \approx \frac{1}{n}\sum_i^n y_i$ for a prediction horizon greater than the value of the kernel length parameter.



(a) Noise free GP distribution

(b) Noisy GP distribution $\sigma^2 = 0.5^2$

Figure 2.14: Noisy and noise free GP distribution

2.5.5 Parameter optimisation

The Gaussian process prior presents possible fit functions within R^n dimensional space as dictated by the inputs and the chosen kernels. To select the best possible functions that could explain the data, the hyperparameters of the underlying model needs to be optimised. In probabilistic models, the search for a function that fits the data is equivalent to evaluating a model that has a higher chance of generating the observe data. As a result, for gaussian process models the likelihood function given in equation (2.38) is used as an objective function for hyperparameter optimisation. This function combines the contribution of the prior, kernels, inputs, observed values and turns it into a single constant number that can be used for parameter selection and model comparison. When the kernel functions are of the same family, parameter optimisation is also synonyms with

model optimisation [89]. In multiple kernel learning, parameter optimisation is associated with model selection. The usual approach in either of these cases, is to optimise the hyperparameters of the kernels using the log-marginal likelihood function. The logarithmic scale is applied only for the sake of mathematical convenience and numerical stability. For instance, given the covariance matrix Σ_{ff}, the parameters $\theta_1, \theta_2, \cdot\cdot, \theta_d$, where $\theta_i \, \epsilon \, \theta$ represent the hyperparameters including the data variance σ^2 and the marginal likelihood function $p(y|f, x, \theta)$,

$$p(y|f, x, \theta) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_{ff} + \sigma^2 I_d|^{1/2}} exp\left(-\tfrac{1}{2} y^T (\Sigma_{ff} + \sigma^2 I_d)^{-1} y\right)$$

Applying logarithmic transformation on LHS and RHS, the log-likelihood is given by

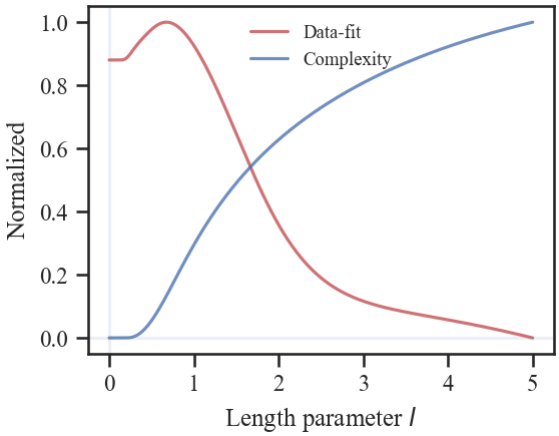
$$\log p(y|f, x, \theta) = -\frac{1}{2} y^T (\Sigma_{ff} + \sigma^2 I_d)^{-1} y - \frac{1}{2} \log |\Sigma_{ff} + \sigma^2 I_d| - \frac{\mathrm{N}}{2} \log(2\pi) \tag{2.41}$$

The log-likelihood function is a differentiable function. However, the presence of the determinant $|\Sigma_{ff} + \sigma^2 I_d|$ term makes it a non-convex function. This in return molds the parameter search procedure into a non-convex optimisation problem. A problem where the optimisation returns a number of feasible regions containing multiple locally optimal values θ_*. Each of these optimal regions corresponds to different model with a particular interpretation of the given data. Hence, this log-likelihood based optimisation doesn't guarantee a global optimum hyperparameter values. Instead, the estimation provides various feasible models with different contextual information in regards to the optimal fit functions. Furthermore, the non-convexity of the objective function puts an exponential time requirement for evaluating feasible optimal values in the event of multiple kernel learning with a large number of parameters. Be that as it may, the Gausssian process models are robust and intuitive. As such, the local minima's are capable of providing a model with acceptable predictive performance. Consequently, the log-likelihood as an objective function and any second-order gradient decent algorithm as an optimiser have been employed to find the optimal parameter values θ_i that maximise the $\log p(y|f, x, \theta)$.

$$\theta_{i_*} = \arg\max_{\theta_i \, \epsilon \, \theta} \left\{ -\frac{1}{2} y^T (\Sigma_{ff} + \sigma^2 I_d)^{-1} y - \frac{1}{2} \log |\Sigma_{ff} + \sigma^2 I_d| - \frac{\mathrm{N}}{2} \log(2\pi) \right\} \tag{2.42}$$

As it can be seen in the above equation, the log-likelihood function is composed of three terms. The first term is associated with the observed data Y. As a result, it is mostly referred as the data-fit. The second term contains the kernel functions, parameters and training points, all the variables that influence model behaviour and performance. It puts a sort of regularization on model complexity or the parameter attainable values. Hence, it is called the complexity term. The third and final term is the normalizing constant, a constant that doesn't affect the optimisation process or value of the optimal parameter. As such, it is mostly ignored during the maximization process.

The complexity term doesn't mention the observed values Y. That means the breadth of the prior distribution, the smoothness of the returned functions or the complexity of the model, have more to do with the inputs, kernels and their hyperparameters values than the actual observation. The balance between the data-fit and complexity term resembles the bias-variance trade off and determines whether the resulting model overfit or under-fit the given problem. For instance, Figure 2.15a shows the evolution of a normalized data-fit and complexity term with respect to the SE kernel length parameter l. As $l \to 0$, the complexity term converges to 0 and the data-fit term converge to a constant C depending on the observed values Y. On the other hand, when $l \to \infty$, the data-fit term decrease and the complexity term increase as shown in Figure 2.15a. In section 2.4.4, we have said, large values of l have a smoothing effect and results in a low order model or lower



(a) A normalized data-fit and complexity term versus the length hyper-parameter l

complexity. However, here the complexity term behave opposite to that assertion only because of the negative in the log-likelihood function in equation (2.41). The complexity term shouldn't be confused with the order of the model. Accordingly, the increase in complexity term as $l \to \infty$, signifies the decrease in model complexity. The data-fit, complexity versus hyperparameter graph differs depending on observed values, inputs and range of the hyperparameter. However, one thing remains constant, when data-fit increases, the complexity decreases and vice versa. As a result, in a similar manner to the bias-variance trade-off, optimal hyperparameter values need to provide a well balanced data-fit values without worsening the complexity term.

The unfortunate consequence of employing the likelihood $p(y|f, x, \theta)$ for parameter optimisation is the associated computational bottlenecks. In equation (2.42), the optimisation algorithm continuously estimate the inverse of the covariance matrix $(\Sigma_{ff}+\sigma^2 I_d)^{-1}$. This requires numerical operations $\mathcal{O}(n^3)$ cubic to the number of observations n. Furthermore, we have to retain the whole data $\{(x_i, y_i)\}_{i=1}^n$ during the optimisation steps and posterior estimation in equation (2.42) and (2.40), which puts a quadratic $\mathcal{O}(n^2)$ memory growth requirements. Consequently, likelihood-based parameter optimisation results in an intensive computation and a higher storage requirement. It is exponentially demanding for big data with a higher dimensional covariance matrix. These realities has made the Gaussian process unscalable and unfit for the analysis of big data [75, 90].

The intractability of the Bayesian inference and the associated computational bottlenecks can be circumvented through the adoption of approximate inference. These techniques can be stochastic like the sparse approximation or they can be deterministic in nature. Methods that follow deterministic approximation focus on drawing inference using the most probable outcome. For instance, the Laplace approximation draws probabilistic inference around the mode of the given distribution [91], while variational based deterministic approximations focus on approximate posterior utilizing the lower bounds of the marginal probability distribution [92]. It is also paramount to point out that not a single approach is successful in approximating all problems regardless of the nature of the approximation. Meaning that, there are problems that either the deterministic or stochastic approach can approximate well and problems in which one of them might not be suitable for. For instance, the Laplace approximation is the easiest approximation. And yet, it might not be the closest Gaussian approximation as compared to those deterministic inferences that employ the KL-divergence as a criteria. On the other hand, the KL-divergence is a good metrics for measuring the discrepancy between two distributions. But, applying the KL-divergence as a measure of similarity to approximate a complex distribution through a simple variational distribution lowers the variance estimation. As such, a successful technique follows a logical path of matching the specific approxima-

tion method with the given problem at hand. However, the common denominator in all approximate inference mechanisms is that their representation comes with a reduced accuracy. Furthermore, employing these methods doesn't entail a reduction in difficulty. In fact, approximate inference is basically a trade off between the inconvenience of solving an intractable integral equation with the difficulty of solving optimization problem. In spit of that, the optimization part is more preferred, because it offers a fast, computationally efficient and optimal approximate solution.

In the case of Gaussian process models, sparse approximation methods like sampling for kernel dimensional reduction and variational inference for posterior approximation rectify these issues by avoiding the Bayesian inference and full rank kernel implementation during parameter learning and posterior estimation. For instance, the sparse variational Gaussian approximation that will be discussed in section 2.6.4, evaluates a sparse approximation of the posterior using fewer samples drawn from the training data [66, 93]. These approximations can be carried out either through a stochastic or deterministic based approximate inferences. In the next section, we will present some of the popular techniques utilized as an approximate inference methodologies in the case of the Gaussian process models.

2.6 Gaussian process approximation

The practical limitation of the exact Gaussian inference can be attributed to the demand for a computational complexity of $\mathcal{O}(n^3)$ and storage capacity $\mathcal{O}(n^2)$ where n is the size of the data [99]. Both requirements create equal impediments when analysing a data with millions of observed values. Even so, the recent improvements in computing power is making the computational requirement irrelevant. However, the storage capacity is still limited and becomes the real challenge when working with huge volume of data. Hence, the question that has been asked by so many is how to make the Gaussian process scalable to big data. In all of the proposed approaches, there is no argument that for the model scalability, the rank of the covariance matrix must be minimized [100]. These approaches argue that rank minimization can be achieved by building the model prior on m pseudo data points that are capable of approximating the observed data rather than building the model relying on the original n observations. By compressing the information content of the gathered data with an $m \ll n$ points, the model's computational and memory requirements can be reduced to the order of the approximating points.

The most important development of this shift in perspective is that the proposed pseudo points may or may not be real data points. Meaning that, they are not necessarily bounded to the domain of the observation. Consequently, they may or may not be part of the training data, as well as no real constraints associated with their exact location. In practice, they have been selected stochastically from the training data, or optimised as part of the fitting process. Hence, there hasn't been a uniform consensus on how they should be selected and what aspect of the model they should approximate. This diverging opinions has given rise to the model-based and posterior-based approximation methods.

Model-based approximations ensure model scalability by approximating the likelihood of the exact Gaussian. Thereby effectively replacing the original model by another model that has a low rank covariance matrix and high computational efficiency. As a result, some aspect of the original model is lost. On the other hand, posterior-based approximation methods such as those that employ the variational free energy (VFE), rely on variational inference and data compression to approximate the posterior distribution while keeping the original GP model intact. In such approximations, the m pseudo points are used to establish a surrogate variational distribution with a sole purpose of approximating the

posterior. To that end, the KL-divergence between the variational and the true posterior is implemented as a metric to quantify the difference, ultimately leading to the evidence lower bound (ELBO) derivation. Here as well, there is no uniform agreement on the procedure for selecting the mathematical form of the approximating (i.e variational) distribution. The choice has largely depended on the specific problem at hand and the desired level of accuracy. This difference has provided a family of posterior-based approximation methods such as the mean-field variational family [94, 95], Gaussian variational family [92, 96], exponential variational family [97, 98], so and so forth.

Despite their difference, something that remained constant in both the posterior and model-based approximation is that the mathematical form of the approximating distribution rely on factorization. However, on which part of the model is the factorization being performed differ from one approach to another. For example, in VFE the factorization is carried out in approximating the posterior distribution of the model. And, this factorization primarily focus on constructing the surrogate or the variational distribution. In case of model-based approximation methods, such as the FITC, since it assumes a full conditional independence of the outputs given the inputs, the joint conditional distribution is given as a factorized approximations of the output distributions.

Among the available model-based approximations, the deterministic training conditional (DTC), the partial independent training conditional (PITC) and the fully independent training conditional (FITC) are the most widely utilized methods for GP model approximation. These methods share similarity on the principle of what part of the model to use for the approximation. However, they differ on the exact strategy followed. For instance, equation (2.33) and equation (2.36) establish the exact GP prior and likelihood as

$$\begin{aligned} p\left(f, f_*\right) &\sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xx_*} \\ \Sigma_{xx_*}^T & \Sigma_{x_*x_*} \end{bmatrix}\right) \\ y &= f + \epsilon \text{ , where, } \epsilon \sim \mathcal{N}(0, \sigma^2 I_d) \\ y &\sim \mathcal{N}(f, \sigma^2 I_d) \end{aligned} \tag{2.43}$$

This derivation assumes a correlation between the latent random variables at the observed points f and all other points f_* as shown by the thick horizontal line connecting f and f_* in the GP graphical model representation in Figure 2.16. Additionally, for the likelihood derivation it assumes a conditional independence among the observed values y given f as shown by the dangling y values in Figure 2.16. This has resulted in a conditional posterior

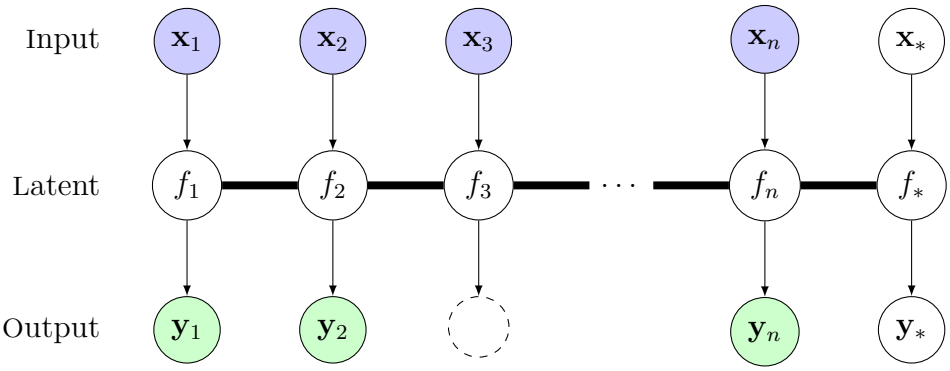


Figure 2.16: The exact GP graphical model: *Missing and observed value are shown with a broken and solid circle respectively. The latent random variables f and f_* are connected by a unbroken horizontal line to signify full correlation. The observed values $y_1, y_2, \cdot\cdot, y_n$ are conditionally independent given f. Hence, they are shown as dangling along with their respective f.*

distribution given in equation (2.40) for the exact GP as

$$p(f_*|y) \sim \mathcal{N}(\Sigma_{x_*x}(\Sigma_{xx} + \sigma^2 I_d)^{-1} y,\ \Sigma_{x_*x_*} - \Sigma_{x_*x}(\Sigma_{xx} + \sigma^2 I_d)^{-1}\Sigma_{xx_*}^T) \tag{2.44}$$

The first theoretical foundation behind the DTC, PITC and FITC approximation is the

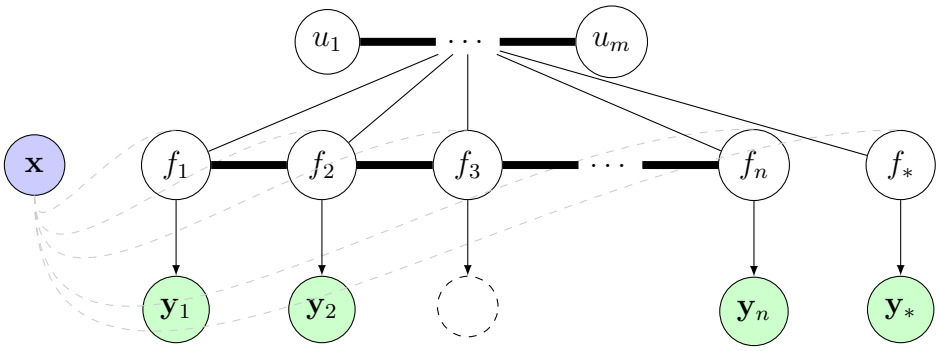


Figure 2.17: Model-based GP approximation: *As the connection between f and f_* is broken, after observing some data y any inference about f_* comes through the inducing variable u. Hence, u serves as a link between the observed data and points at the forecast horizon.*

assumption of a conditional independence between the latent variables at training (x, f) and test points (x_*, f_*) given the random variables at pseudo points (z_m, u_m). Such assumption effectively breaks the correlation link between the latent variables at the training point f from those at forecast f_* as shown by the absence of a connection line between f and f_* in Figure 2.17. As such, any impact the observed data has on the forecast point is induced indirectly through the pseudo variables u_m. As a result, the random variables representing the pseudo points are also commonly referred as the inducing random variables. There are various researches on model-based approximations. Some with different names and yet with a similar methods of implementation as discussed in detail [99, 100, 101]. The general framework to sparsify the covariance matrix into creating a computationally feasible model through model-based approximations starts by augmenting the exact GP with few inducing random variables. For instance, assuming we have $\{u_i\}_{i=1}^m$ inducing variables, the augmented joint distribution $p(f, f_*, u)$ can be written as

$$p\left(f, f_*, u\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{uu} & \Sigma_{uf} & \Sigma_{uf_*} \\ \Sigma_{fu} & \Sigma_{ff} & \Sigma_{ff_*} \\ \Sigma_{f_*u} & \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}\right) \tag{2.45}$$

where $p(u) \sim \mathcal{N}(0, \Sigma_{uu})$ is the prior on the inducing variables. From equation (2.45), we can estimate the conditional distribution of f and f_* given u

$$\begin{aligned} p(f|u) &\sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}) \\ p(f_*|u) &\sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}u\ ,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf_*}) \end{aligned} \tag{2.46}$$

These results resemble exactly the posterior distribution for the exact GP. The conditional distribution $p(f|u)$ in equation (2.46) still contains the original matrix Σ_{ff}. Such representation wouldn't help us in improving the computational efficiency of the model. Hence, the objective in all sparse approximation is to estimate the conditionals $p(f|u)$ and $p(f_*|u)$ in a way that minimize or eliminate the contribution of Σ_{ff}. The different approaches and assumptions made to realize an approximate estimate to these conditionals has given rise to a number of model-based GP approximation methods.

2.6.1 Deterministic training conditional (DTC) approximation

The DTC is a model-based approximation method that has been also called the projected latent variable method and projected process approximation method in different literature [101]. The DTC approximate formulation rests on two important consideration. First, it assumes that there is a deterministic relation between the inducing random variable u and the latent random variable f with zero variance. Consequently, it provides an approximate conditional for $p(f|u)$ given in equation (2.46)

$$p(f|u) \approx q(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ 0) \tag{2.47}$$

This deterministic assumption ignores the stochastic nature of the covariance among f & u. In doing so, it completely avoids the computational bottleneck associated with covariance matrix Σ_{ff}. The deterministic mapping goes further into providing an approximate likelihood $q(y|f)$,

$$\begin{aligned} p(y|f) \approx q(y|f) &\sim \mathcal{N}(f, \sigma^2 I_n) \\ &\sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \sigma^2 I_n) \end{aligned} \tag{2.48}$$

Marginalizing $u \sim \mathcal{N}(0, \Sigma_{uu})$, the DTC approximated likelihood $q(y|f)$ is given by

$$q(y|f) = \mathcal{N}(0\ , \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \sigma^2 I_n) \tag{2.49}$$

Because of this DTC is also referenced as a likelihood-based approximation method. The second consideration is related to how best to approximate the conditional $p(f_*|u)$. We could follow the same approach to approximate $p(f_*|u)$, as we did for $p(f|u)$. However, considering a deterministic behaviour for the posterior distribution can constrain uncertainties propagation [101]. To ensure variance propagation during prediction, DTC allows f_* to keep its prior variance $\Sigma_{f_*f_*}$. As a result, the approximate conditional $q(f_*|u)$ retains the same conditional distribution for $f_* \Rightarrow q(f_*|u) = p(f_*|u)$. Given the approximated conditional $q(f|u)$ and $q(f_*|u)$, we can establish the prior distribution for f and f_* under the DTC method. Applying the linear Gaussian transformation on $q(f|u)$ and $q(f_*|u)$

$$\begin{aligned} q(f|u) &\sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ 0) \ \text{ where } u \sim \mathcal{N}(0, \Sigma_{uu}) \\ q(f) &\sim \mathcal{N}(0\ ,\ \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{fu}^{T}) \ \text{ and} \\ q(f_*|u) &= p(f_*|u) \\ q(f_*|u) &\sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}u\ ,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf_*}) \\ q(f_*) &\sim \mathcal{N}(0\ ,\ \Sigma_{f_*f_*}) \end{aligned} \tag{2.50}$$

Now that we have $q(f)$ and $q(f_*)$, the DTC approximated prior can be defined

$$\begin{aligned} q(f, f_*) &\sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} & cov(f, f_*) \\ cov(f, f_*) & \Sigma_{f_*f_*} \end{bmatrix}\right) \\ cov(f, f_*) &= \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{f_*u} \\ q(f, f_*) &\sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} & \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf_*} \\ \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf} & \Sigma_{f_*f_*} \end{bmatrix}\right) \end{aligned} \tag{2.51}$$

Given the approximated likelihood $q(y|f) \sim \mathcal{N}(0\ , \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \sigma^2 I_n)$, and assigning $\Lambda = \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \sigma^2 I_n$, the DTC approximated posterior can be estimated as

$$q(f_*|y) \sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf}\Lambda^{-1}y\ ,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf}\Lambda^{-1}\Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf_*}) \tag{2.52}$$

The DTC decision to retain the prior variance $\Sigma_{f_*f_*}$ at the forecast points has created inconsistency on variance evaluation. The variance estimation at a given point varies

depending on whether that point is part of the training data or not. The graphical representation of the DTC approximation is shown in Figure 2.18. Due to the conditional independence of the latent variables f and f_* given u, all communication path have been removed. However, f_* is allowed to retain its prior to ensure the propagation of uncertainties during prediction. Meaning that the elements of f_* may or may not retains their mutual interdependence depending on the nature of $\Sigma_{f_*f_*}$ as shown by the thick connection line between the elements of f_* in Figure 2.18. If matrix $\Sigma_{f_*f_*}$ is diagonal then the horizontal line connecting the elements of f_* can be ignored.

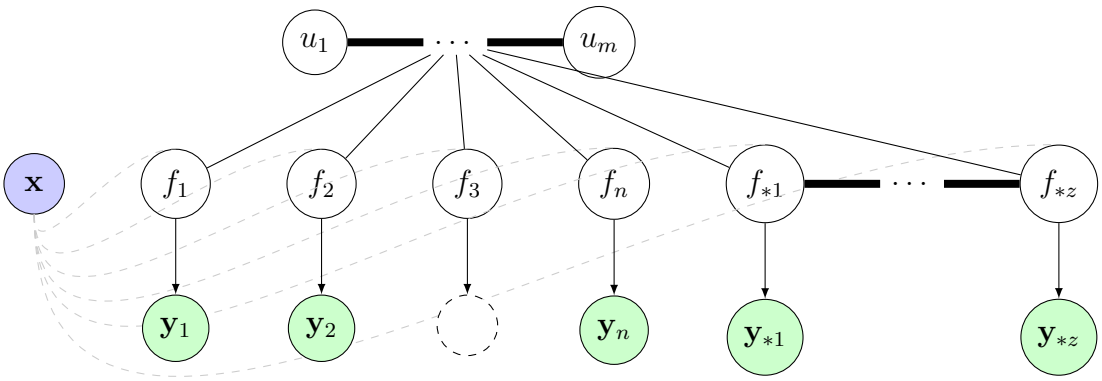


Figure 2.18: Graphical model for DTC approximation: *the random variables f and f_* are assumed to be conditionally independent given u. As such, all paths between the latent variables have been severed.*

2.6.2 Fully independent training conditional (FITC) approx.

The DTC answer to the issue of scalability and computational inefficiencies in the exact GP inference was to completely remove the covariance $\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$ at training points from the conditional $p(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf})$. And, provided an approximate conditional $q(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ 0)$ in its place. However, such approximation has made the relationship deterministic and constrained the rich interactions between the inducing and latent variables. The computational cost associated with inverting the matrix $\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$ is understandable. And yet, it can be made manageable by turning it into a diagonal matrix. That is precisely the idea behind the fully independent training conditional approximation. The FITC proposes an alternative approximation $q(f|u)$ to the conditional $p(f|u)$ that provides a robust interaction between the inducing and latent variables while keeping the computational cost minimum. For instance, given the conditional $p(f|u)$

$$p(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}) \tag{2.53}$$

and assuming full independence between the latent variables or alternatively considering the diagonal part of the covariance matrix, the FITC approximated conditional $q(f|u)$ is given by

$$p(f|u) \approx q(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \mathrm{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}]) \tag{2.54}$$

The size of the matrix $\mathrm{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}]$ extends to the full range of the training data. In return this results in a more flexible likelihood approximation $q(y|f)$

$$\begin{aligned} p(y|f) \approx q(y|f) &\sim \mathcal{N}(f, \sigma^2 I_n) \\ &\sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u\ ,\ \mathrm{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n) \end{aligned} \tag{2.55}$$

Marginalizing $u \sim \mathcal{N}(0, \Sigma_{uu})$, the FITC approximated likelihood $q(y|f)$ is given by

$$q(y|f) \sim \mathcal{N}(0,\ \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n) \tag{2.56}$$

Except for the addition of the covariance $\text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}]$, the derivation of FITC is similar to the DTC approximation. As such, just like in case of DTC, the FITC allows the conditional at forecast point $q(f_*|u)$ to retain its prior variance $\Rightarrow q(f_*|u) \approx p(f_*|u)$. Consequently, the approximated prior in the case of FITC is given by

$$q(f, f_*) \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] & \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf_*} \\ \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf} & \Sigma_{f_*f_*} \end{bmatrix}\right) \tag{2.57}$$

As we have said before the difference between DTC and FITC is the addition of the diagonal matrix. Hence, assuming $\Lambda = \text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n$, and $\Sigma_q = \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$, the FITC approximated predictive distribution $q(f_*|y)$ is given by

$$q(f_*|y) \sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf}(\Sigma_q + \Lambda)^{-1}y\ ,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf}(\Sigma_q + \Lambda)^{-1}\Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf_*}) \tag{2.58}$$

The graphical model of the FITC is similar to the DTC and is shown in Figure 2.18. In the case of DTC, conditional independence between the latent variables or the absence of information path between the latent variables in Figure 2.18, is guaranteed by removing the correlation matrix in driving the approximated conditional. In the case of FITC, conditional independence is ensured by diagonalizing the correlation matrix which in principle breaks the information path as shown in Figure 2.18. However, the nature of interaction between the latent variables at forecast point is truly independent if and only if the matrix $\Sigma_{f_*f_*}$ is diagonal. If that happens the connection line between the elements of f_* in Figure 2.18 should be omitted.

2.6.3 Optimal inducing locations

Modeling through pseudo points creates an opportunity to take advantage of the redundancy and correlation that might exist in the input data so that a model defined in a minor subspace can have an equivalent inference capacity compared to a model defined in the entirety of the input space. And yet, the gained reduction in computational load is paid with the loss of predictive accuracy. No matter how good the approximation, the resulting model will not be as accurate as a model trained on a full data. Regardless, the trade off can be made affordable through a careful selection of the placement and number of the pseudo points. Fixing the number of the inducing variables is a matter of tuning based on the desired level of accuracy and the computational load that one is willing to bear for the task at hand. However, the determination of the optimal inducing points is more of an optimization problem. Good inducing locations that are capable of representing the information content of the whole data set requires an iterative procedure. One effective approach that is suggested is through the maximization of the marginal likelihood.

$$\zeta_{i_*} = \arg\max_{\zeta_i\ \epsilon\ \zeta} \Big\{ \log p(y|z, u, \theta) \Big\} \tag{2.59}$$

where $\zeta_i = \{z_i, u_i, \theta_i\}$ represent the inducing location, inducing value and the kernel hyperparameters respectively. Employing the likelihood as a criterion allows the optimisation of both the hyperparameters and the inducing variables to be made in one go. In regards to the pseudo points learning, the brute force approach to their optimisation will be to

maximise the likelihood with respect z and u. However, the number of parameters to optimise can be lowered by following a more economical procedure. One approach that is suggested in [102] is to place a Gaussian prior on the inducing values u and marginalize it out of the likelihood definition. This approach is inline with the first assumption placed upon the pseudo points. It is assumed that there is no constraint regarding pseudo point location. The only requirement if any, that is placed upon the inducing variables is that they should be jointly distributed with the latent random variable f, f_* of the original data if they are to approximate it well. Consequently, placing a Gaussian prior on the inducing variables u

$$p(u|z) \sim \mathcal{N}(0, \Sigma_{uu}) \tag{2.60}$$

we have derived the DTC and FITC approximated likelihood in equation (2.49) and (2.56) as

$$q(y|f) \sim \mathcal{N}(0,\ \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \Lambda) \tag{2.61}$$

where $\Lambda = \sigma^2 I_n$ for DTC and $\Lambda = \text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n$ in case of FITC. The marginalization of u from the likelihood definition relieves the computational cost associated with its evaluation so that the maximization is more focused on its location z. As such, the optimal inducing locations are estimated through the maximisation of the likelihood $\log q(y|z,\theta)$ given in equation (2.61)

$$\log q(y|z,\theta) = -\frac{1}{2}y^T(\Sigma_q + \Lambda)^{-1}y - \frac{1}{2}\log|\Sigma_q + \Lambda| - \frac{n}{2}\log(2\pi) \tag{2.62}$$

where n is the size of the training data, $\Sigma_q = \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$, $\Lambda = \sigma^2 I_n$ for DTC and $\Lambda = \text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n$ for FITC. The likelihood criterion continuously evaluates the inverse $(\Sigma_q + \Lambda)^{-1}$ and the determinant $|\Sigma_q + \Lambda|$ during the optimisation. In its current form, the covariance matrix $\Sigma_q + \Lambda$ is still an n x n matrix which makes the estimation of the inverse and determinate infeasible for large n. Fortunately, the nature of the covariance matrix permits alternative matrix representation through the woodbury matrix identity and woodbury determinant lemma [103]. The woodbury matrix identity states that given a n x n matrix Σ such that

$$\Sigma = \text{A}^{-1} + \text{X}\text{B}^{-1}\text{X}^T \tag{2.63}$$

where A,X and B are conformable matrices with size n x n, n x m and m x m respectively. Then its inverse Σ^{-1} and determinant $|\,\Sigma\,|$ can be evaluated as

$$\begin{aligned} \Sigma^{-1} &= \text{A} - \text{A}\text{X}\text{P}^{-1}\text{X}^T\text{A} \\ |\,\Sigma\,| &= |\text{P}||\text{A}|^{-1}|\text{B}|^{-1} \end{aligned} \tag{2.64}$$

where the matrix $\text{P} = \text{B} + \text{X}^T\text{A}\text{X}$ is an m x m matrix. Hence, in the DTC likelihood approximation, the inverse and determinant of the covariance matrix $(\Sigma_q + \Lambda) = \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \sigma^2 I_n \Rightarrow (\sigma^{-2}I_n)^{-1} + \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$ can be conveniently evaluated using the two woodbury relationships by assigning $\text{A} = \sigma^{-2}I_n$, $\text{B} = \Sigma_{uu}$, $\text{X} = \Sigma_{fu}$ and $\text{P} = \Sigma_{uu} + \sigma^{-2}\Sigma_{uf}\Sigma_{fu}$. In a similar manner for the FITC likelihood approximation, by assigning $\text{A} = (\text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n)^{-1}$, $\text{B} = \Sigma_{uu}$, $\text{X} = \Sigma_{fu}$ and $\text{P} = \Sigma_{uu} + \Sigma_{uf}(\text{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n)^{-1}\Sigma_{fu}$ computational efficiency can be improved. In both cases the matrix A is an n x n diagonal matrix. As such, determining A^{-1} is computationally feasible.

The covariance matrices in the approximate posteriors $q(f_*|y)$ given in equation (2.52) and (2.58) are still n x n. Analogous to the log-likelihood formulation, the same principle can be applied in rewriting the DTC and FITC approximate posterior into a more computationally convenient form. For instance, substituting the values of the matrix A,

B, X, P that is used in the log-likelihood into the DTC approximate posterior $q(f_*|y)$ and simplifying it, $q(f_*|y)$ can be rewritten into a form that has the inverse of m x m matrix, where m is the size of the inducing variables

$$q(f_*|y) \sim \mathcal{N}(\Sigma_{f_*u}\mathrm{P}^{-1}\mathrm{X}^T\mathrm{A}y, \Sigma_{f_*f_*} - \Sigma_{f_*u}\mathrm{B}^{-1}\Sigma_{uf_*} + \Sigma_{f_*u}\mathrm{P}^{-1}\Sigma_{uf_*}) \tag{2.65}$$

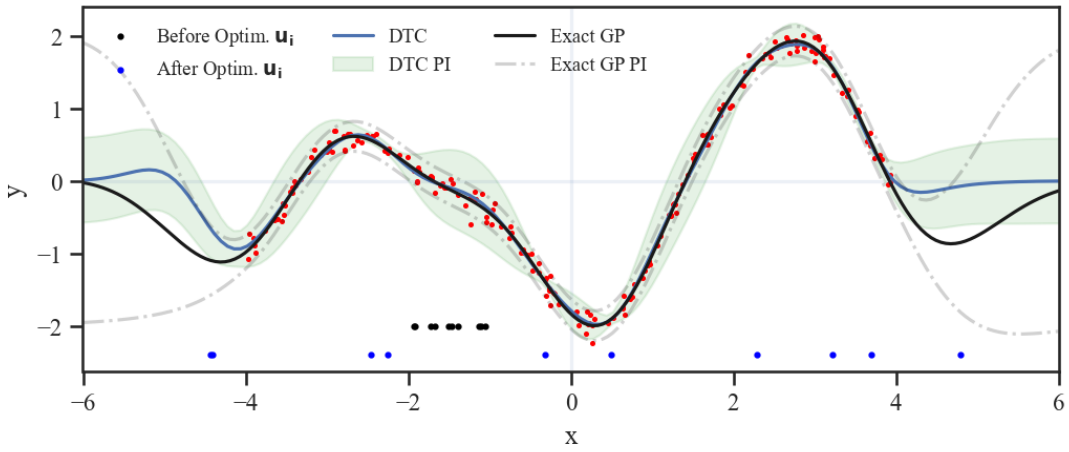
where $\mathrm{P} = \Sigma_{uu} + \sigma^{-2}\Sigma_{uf}\Sigma_{fu}$, $\mathrm{B} = \Sigma_{uu}$, $\mathrm{A} = \sigma^{-2}I_n$ and $\mathrm{X} = \Sigma_{fu}$. On the other hand, the estimate for FITC approximate posterior will have a similar form to the posterior given in equation (2.65), except for a minor change in the matrix $\mathrm{A} = (\mathrm{diag}[\Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}] + \sigma^2 I_n)^{-1}$ and $\mathrm{P} = \Sigma_{uu} + \Sigma_{uf}A\Sigma_{fu}$.

The impact of pseudo point inclusion on data compression and model approximation is evident in the DTC and FITC predictive distribution shown in Figure 2.19. We applied the model approximations on a randomly generated function. The squared exponential kernel and ten inducing locations were selected to map and approximate the covariance matrix. In regards to tracking the mean trajectory, both DTC and FITC deliver acceptable model approximations based on how close the inducing locations are to the forecast point. In fact, the FITC model tends to overfit the given observations. However, the estimated variance varies depending on the spread of the inducing points. The additional diagonal component on the covariance matrix of FITC makes the model to underestimate the variance in densely populated inducing areas. Thereby, affecting its coverage probability compared to the DTC and the exact GP as shown in Figure 2.19a and Figure 2.19b. In spite of that, both models ensure variance propagation and for forecast point that are far from the inducing locations, the estimated variance will approach the prior variance $\Sigma_{f_*f_*}$ (i.e $\Sigma_{f_*u}\mathrm{B}^{-1}\Sigma_{uf_*} + \Sigma_{f_*u}\mathrm{P}^{-1}\Sigma_{uf_*} \to 0$).

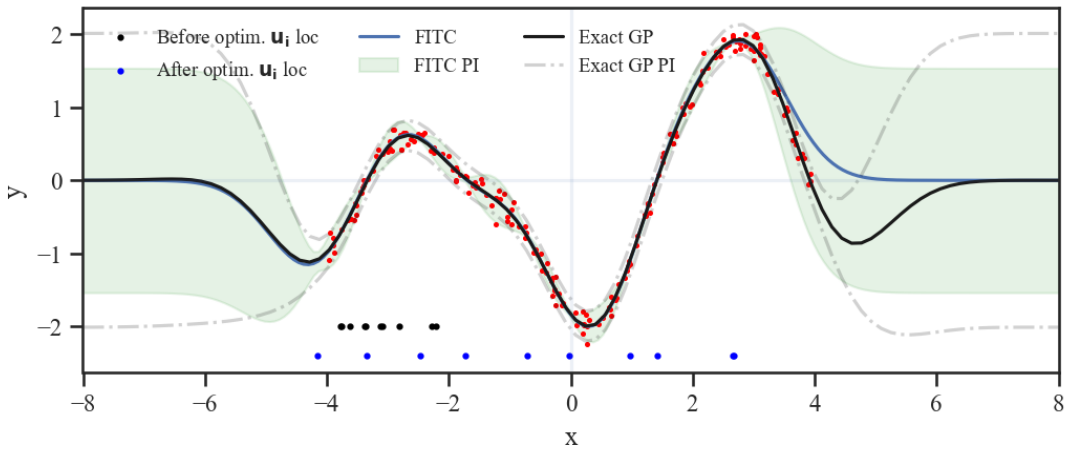
In general, the apparent effect of this diverging approach to Gaussian process approximation is that the model-based approximation methods such as FITC and DTC result in a faster computation where as posterior-based approximation like the VFE are computationally intensive compared to FITC or DTC. However, VEF allow complex covariance to be defined among the latent random variables which is not possible due to the independence assumption in model-based approximations. Although model-based approximations shows a fast computational efficiency compared to the VFE, they run into the risk of overfitting when optimizing the model parameters. This is due to the result of working on approximating the model itself. However, in the case of the VFE, the hyper-parameter optimization is carried out on the lower bounds of the marginal likelihood. Meaning that the true marginal likelihood is always greater than the ELBO. As such, the optimized parameters will not overfit the model worse than any parameters optimized using the true marginal likelihood. In section 2.6.4, we present the variational based Gaussian approximation.

2.6.4 Sparse variational gaussian process (SVGP) approximation

In the previous section we discussed the DTC and FITC model based approximation methods. And, we have said rank minimization for the covariance matrix can be carried out by utilizing extra pseudo points. The approach that is followed in approximating the aspect of the model through these pseudo points, determines the nature and predictive accuracy of the resulting model. Like the DTC and FITC, the sparse variational Gaussian process (SVGP) is another popular approximation method based on pseudo points. However, unlike the previous, SVGP is a method that try to ensure model scalability using a variational based approximate posterior. It follows sparse modeling and variational inference to build a computationally efficient inferential model. To that end, it utilize an auxiliary distribution with the objective of optimising its parameters until its divergence from the true posterior is minimised. This surrogate distribution is commonly called the



(a) Exact GP Vs DTC approximate predictive distribution



(b) Exact GP Vs FITC approximate predictive distribution

Figure 2.19: Model-based GP approximate methods: *The dark and blue points show the inducing variables* $\mathbf{u}_i$ *locations before and after optimisation. Although, 10 inducing locations were used for the approximation, only 9 on them resides within the training range.*

variational distribution [71]. The characteristics of this distribution is determined by the placement of the pseudo points. Similar to the model-based approximation methods, the SVGP models place the same requirements on the pseudo-points. No constraints on their placement and the expectation of a joint distribution with the latent variables f and f_* of the original data. However, the diverging approaches in regards to the form of the variational distribution has resulted in a number of posterior-based approximation methods. Such as, the mean-field and Gaussian variational family distributions mentioned in section 2.6. However, in this section we follow the approach outlined in [66]. Before diving into the SVGP derivation, lets review the step that lead to the computational bottleneck in the exact GP inference. In equation (2.33), we established the prior distribution and the likelihood of the exact GP model

$$\begin{aligned} p\left(f, f_*\right) &\sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{ff} & \Sigma_{ff_*} \\ \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}\right) \\ y &= f + \epsilon \text{ , where, } \epsilon \sim \mathcal{N}(0, \sigma^2 I_d) \\ y &\sim \mathcal{N}(f, \sigma^2 I_d) \end{aligned} \tag{2.66}$$

with a true posterior joint distribution

$$p(y, f) = p(y|f)p(f) \tag{2.67}$$

where $p(f) \sim \mathcal{N}(0, \Sigma_{ff})$. If we apply Bayesian inference on equation (2.66) - (2.67), it will eventually lead us to the exact GP with the expensive covariance matrix $\Sigma_{ff} + \sigma^2 I_n$. For an alternative parameter learning and forecast derivation, the sparse variational method makes two important arguments. First, it assumes that for a given data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, we can find m (i.e where $m \ll n$) inducing locations $\{z_i\}_{i=1}^m$ that can be used to summarize the data. Good data summerization entails that the inducing random variables $\{u_i\}_{i=1}^m$ at these locations should be jointly distributed with the latent variables f and f_*. Furthermore, it assumes that the latent variables of the original data are conditionally independent given the inducing variable $u \Rightarrow p(f_*|f, u) = p(f_*|u)$. Consequently, it defines an augmented sparse GP prior

$$p\left(f, f_*, u\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{uu} & \Sigma_{uf} & \Sigma_{uf_*} \\ \Sigma_{fu} & \Sigma_{ff} & \Sigma_{ff_*} \\ \Sigma_{f_*u} & \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}\right) \tag{2.68}$$

with augmented true posterior joint distribution

$$\begin{aligned} p(y, f, u) &= p(y|f, u)p(f|u)p(u) \\ &= p(y|f)p(f|u)p(u) \end{aligned} \tag{2.69}$$

Despite the augmentation, the application of Bayesian inference on equation (2.69) will marginalize and remove the inducing variable u from equation (2.69) which turns it into equation (2.67). Hence, the objective in sparse variational approximation is how to evaluate the augmented joint distribution $p(y, f, u)$ in a way that circumvent the Bayesian inference. To that end, it proposes an augmented variational distribution $q(f, u)$ whose parameters will be optimised through variational inference to approximate the real posterior $p(y, f, u)$ [66]. Consequently, a factorized joint variational distribution $q(f, u)$ is proposed

$$p(y, f, u) \approx q(f, u) = p(f|u)q(u) \tag{2.70}$$

where $p(f|u)$ is the conditional distribution of f from the sparse prior in equation (2.68) and is given by

$$p(f|u) \sim \mathcal{N}(\Sigma_{fu}\Sigma_{uu}^{-1}u,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}) \tag{2.71}$$

and $q(u) = \mathcal{N}(u; \mu_q, \Sigma_q)$, is a new user defined variational distribution drawn from a family of gaussian distribution $\mathcal{Q}$ that will be selected by optimizing the free variational parameter μ_q and Σ_q. Now, from equation (2.70) the marginal distribution of $q(f)$ can be computed as

$$\begin{aligned} q(f) &= \int q(f, u)du \Rightarrow q(f) = \int p(f|u)q(u)du \\ &= \int \mathcal{N}(f \mid \Sigma_{fu}\Sigma_{uu}^{-1}\, u,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf})\mathcal{N}(u|\mu_q, \Sigma_q)du \\ q(f) &\sim \mathcal{N}(f \mid \Sigma_{fu}\Sigma_{uu}^{-1}\,\mu_q,\ \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \Sigma_{fu}\Sigma_{uu}^{-1}\,\Sigma_q(\Sigma_{fu}\Sigma_{uu}^{-1})^T) \\ q(f) &\sim \mathcal{N}(\mathrm{A}, \mathrm{B}) \end{aligned} \tag{2.72}$$

where $\mathrm{A} = \Sigma_{fu}\Sigma_{uu}^{-1}\mu_q$ and $\mathrm{B} = \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf} + \Sigma_{fu}\Sigma_{uu}^{-1}\,\Sigma_q(\Sigma_{fu}\Sigma_{uu}^{-1})^T$. The similarity between the two joint distribution in equation (2.69) and (2.70) is measured and framed as an optimization problem using $\mathrm{KL}(p(f, u, y)||q(f, u))$ divergence, where

$$kl(p(f, u, y)||q(f, u)) = \iint q(f, u)log\frac{q(f, u)}{p(f, u, y)}dfdu \tag{2.73}$$

Substituting equation (2.69) & (2.70) into equation (2.73) and simplifying it

$$\begin{aligned} kl(p(f,u,y)||q(f,u)) &= \iint q(f,u)\log\frac{p(f|u)q(u)}{p(y|f)p(f|u)p(u)}dfdu \\ &= \iint q(f,u)\log\frac{q(u)}{p(y|f)p(u)}dfdu \end{aligned} \tag{2.74}$$

Then the main task in the variational inference would be to seek the parameters that minimize the KL divergence between $p(f,u,y)$ and $q(f,u)$. As a result, during parameter optimisation, equation (2.74) will be minimized with respect to the hyperparameters θ, the inducing locations z.

$$\arg\min_{z,\theta_i}\Big\{kl(p(f,u,y)||q(f,u))\Big\} = \arg\min_{z,\theta_i}\Big\{\iint q(f,u)\log\frac{q(u)}{p(y|f)p(u)}dfdu\Big\} \tag{2.75}$$

However, the KL minimization problem in equation (2.75) can be forwarded as a maximisation problem

$$\begin{aligned} \arg\min_{z,\theta_i}\Big\{kl(p(f,u,y)||q(f,u))\Big\} &= -\Big\{\arg\max_{z,\theta_i}\Big\{\iint q(f,u)\log\frac{q(u)}{p(y|f)p(u)}dfdu\Big\}\Big\} \\ &= \arg\max_{z,\theta_i}\Big\{\iint q(f,u)\log\frac{p(y|f)p(u)}{q(u)}dfdu\Big\} \\ &= \arg\max_{z,\theta_i}\Big\{\iint q(f,u)\log p(y|f)dfdu + \iint q(f,u)\log\frac{p(u)}{q(u)}dfdu\Big\} \\ &= \arg\max_{z,\theta_i}\Big\{\int q(f)\log p(y|f)df + \int q(u)\log\frac{p(u)}{q(u)}du\Big\} \\ &= \arg\max_{z,\theta_i}\Big\{\int q(f)\log p(y|f)df - \int q(u)\log\frac{q(u)}{p(u)}du\Big\} \\ &= \arg\max_{z,\theta_i}\Big\{\int q(f)\log p(y|f)df - kl(q(u)||p(u))\Big\} \end{aligned} \tag{2.76}$$

Equation (2.76) establishes the famous evidence lower bound (ELBO equation.

$$\text{ELBO} = \int q(f)\log p(y|f)df - kl(q(u)||p(u)) \tag{2.77}$$

As such, minimizing $\text{KL}(p(f,u,y)||q(f,u))$ is equivalent to maximizing the ELBO. Hence, through the maximisation of the ELBO, the optimal variational parameters can be evaluated. Assuming the observed values are sampled from a random variable y generated according to

$$\begin{aligned} y &\sim \mathcal{N}(f,\sigma^2 I_n), \text{ then} \\ \log p(y\mid f) &= -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y-f)^T(y-f) \end{aligned} \tag{2.78}$$

It should be noted that in equation (2.78), the assumed Gaussian data distribution is only out of mathematical convenience so that the resulting variational distribution can be compared with the Bayesian version. In fact, the variational methods permits the random variable y to be drawn from any stochastic process regardless of the nature of the error distribution. As such, it doesn't require the likelihood distribution to be strictly normal. In equation (2.77), the likelihood term is an expectation with respect to the variational

distribution $q(f)$. As such, substituting equation (2.72) & (2.78) into equation (2.77) for $q(f)$ and $\log p(y|f)$, the ELBO can be rewritten as

$$\begin{aligned}
\text{ELBO} &= \int q(f) \log p(y|f) df - kl(q(u)||p(u)) \\
&= \mathbb{E}_{q(f)}\Big\{ \log p(y|f) \Big\} - kl(q(u)||p(u)) \\
&= \int \Big\{ -\frac{\text{N}}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y-f)^T(y-f) \Big\} \mathcal{N}(f \mid \text{C,D}) df - kl(q(u)||p(u)) \\
&= \int \Big\{ -\frac{\text{N}}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} tr(yy^T - 2yf^T + f^Tf) \Big\} \mathcal{N}(f \mid \text{C, D}) df - kl(q(u)||p(u)) \\
&= \Big\{ -\frac{\text{N}}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} tr(yy^T - 2y\text{C} + \text{C}^T\text{C} + \text{D}) \Big\} - kl(q(u)||p(u)) \\
&= \Big\{ -\frac{\text{N}}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y-\text{C})^T(y-\text{C}) - \frac{1}{2\sigma^2} tr(\text{D}) \Big\} - kl(q(u)||p(u)) \\
&= \Big\{ \log \mathcal{N}(y \,|\text{C}, \sigma^2 I_n) - \frac{1}{2\sigma^2} tr(\text{D}) \Big\} - kl(q(u)||p(u))
\end{aligned} \tag{2.79}$$

where the $p(y|f) \sim \mathcal{N}(y \,|\text{C}, \sigma^2 I_n)$ and $\text{C} = \Sigma_{fu}\Sigma_{uu}^{-1}u$ & $\text{D} = \Sigma_{ff} - \Sigma_{fu}\Sigma_{uu}^{-1}\Sigma_{uf}$ are given in equation (2.71). For two multivariate gaussian distribution $q(u)$ & $p(u)$, the KL-divergence $kl(q(u)||p(u))$ is given as

$$kl(q(u)||p(u)) = \frac{1}{2}\Big\{ \log \frac{|\Sigma_{uu}|}{|\Sigma_q|} - m + tr(\Sigma_{uu}^{-1}\Sigma_q) + (0-\mu_q)^T\Sigma_{uu}^{-1}(0-\mu_q) \Big\} \tag{2.80}$$

Substituting equation (2.80) in place of $kl(q(u)||p(u))$ in equation (2.79), the analytical form of the ELBO used for parameter optimisation is given by

$$\begin{aligned}
\text{ELBO} = &\Big\{ \log \mathcal{N}(y \,|C, \sigma^2 I_n) - \frac{1}{2\sigma^2} tr(D) \Big\} \\
&- \frac{1}{2}\Big\{ \log \frac{|\Sigma_{uu}|}{|\Sigma_q|} - m + tr(\Sigma_{uu}^{-1}\Sigma_q) + (0-\mu_q)^T\Sigma_{uu}^{-1}(0-\mu_q) \Big\}
\end{aligned} \tag{2.81}$$

The addition of the trace component $\frac{1}{2\sigma^2} tr(\text{D})$ in the ELBO, adds another extra layer of regularization compared to the DTC and FITC log-likelihood in equation (2.62) that is used for parameter learning. As a result, the SVGP is less susceptible to overfitting compared to either of the model based approximations. The KL-divergence between $p(y, f, u)$ & $q(f, u)$ can also be used to evaluate the analytical form of the optimal variational distribution $q^*(u)$. For instance, in equation (2.76), the KL-divergence minimization is represented as a maximisation problem

$$\begin{aligned}
\underset{z,\mu_q,\Sigma_q,\theta_i}{\arg\min} \Big\{ kl(p(f,u,y)||q(f,u)) \Big\} = &\underset{z,\mu_q,\Sigma_q,\theta_i}{\arg\max} \Big\{ \iint q(f,u) \log p(y|f) df du \\
&+ \iint q(f,u) \log \frac{p(u)}{q(u)} df du \Big\}
\end{aligned} \tag{2.82}$$

Hence, the optimal variational distribution $q^*(u)$ is the distribution that minimize the LHS of equation (2.82) or that maximise the RHS of the same equation. Marginalizing f

out of equation (2.82) and then derivating the resulting equation with respect to $q^*(u)$

$$\begin{aligned}
kl(p(f,u,y)||q(f,u)) &= \int q(u)\log p(y|f)du + \int q(u)\log\frac{p(u)}{q(u)}du \\
\frac{\partial}{\partial q^*(u)}kl(p(f,u,y)||q(f,u)) &= \frac{\partial}{\partial q^*(u)}\Big\{\int q(u)\log p(y|f)du + \int q(u)\log\frac{p(u)}{q(u)}du\Big\} \\
\frac{\partial}{\partial q^*(u)}kl(p(f,u,y)||q(f,u)) &= \int\Big\{\log p(y|f) + \log p(u) - \log q(u) - 1\Big\}du \\
0 &= \log p(y|f) + \log p(u) - \log q(u) - 1 \\
\log q(u) &= \log p(y|f)p(u) - 1 \\
\log q(u) &\propto \log p(y|f)p(u)
\end{aligned} \tag{2.83}$$

Due to the proportionality, the optimal variational parameters can be computed through moment matching. From equation (2.79) we have $p(y|f) \sim \mathcal{N}(y\,|\mathrm{C}, \sigma^2 I_n)$ where $\mathrm{C} = (\Sigma_{fu}\Sigma_{uu}^{-1})u$, for the sake of brevity lets represent $\mathrm{C}{=}\mathrm{R}u$ where $\mathrm{R} = \Sigma_{fu}\Sigma_{uu}^{-1}$. From the sparse prior in equation (2.68) we have $p(u) \sim \mathcal{N}(0, \Sigma_{uu})$, substituting these in their respective place in equation (2.83) and applying moment matching

$$\begin{aligned}
\log q(u) &\propto \log p(y|f)p(u) \\
\log\mathcal{N}(\mu_q^*, \Sigma_q^*) &\propto \log\Big\{\mathcal{N}(y\,|\mathrm{C}, \sigma^2 I_n)\mathcal{N}(0, \Sigma_{uu})\Big\} \\
-\frac{1}{2}(u-\mu_q^*)^T\Sigma_q^{*-1}(u-\mu_q^*) &\propto -\frac{1}{2\sigma^2}(y-\mathrm{C})^T(y-\mathrm{C}) - \frac{1}{2}u^T\Sigma_{uu}^{-1}u \\
-\frac{1}{2}(u-\mu_q^*)^T\Sigma_q^{*-1}(u-\mu_q^*) &\propto -\frac{1}{2\sigma^2}(y-\mathrm{R}u)^T(y-\mathrm{R}u) - \frac{1}{2}u^T\Sigma_{uu}^{-1}u \\
-\frac{1}{2}(u^T\Sigma_q^{*-1}u - 2u^T\Sigma_q^{*-1}\mu_q^* + \mu_q^{*T}\mu_q^*) &\propto -\frac{1}{2\sigma^2}(y^Ty - 2u^T\mathrm{R}^Ty + u^T(\mathrm{R}^T\mathrm{R})u) - \frac{1}{2}u^T\Sigma_{uu}^{-1}u \\
-\frac{1}{2}(u^T{\Sigma^*}_q^{-1}u - 2u^T\Sigma_q^{*-1}\mu_q^*) &\propto -\frac{1}{2}(u^T(\frac{\mathrm{R}^T\mathrm{R}}{\sigma^2} + \Sigma_{uu}^{-1})u - 2u^T(\frac{\mathrm{R}^Ty}{\sigma^2})) \\
\Sigma_q^{*-1} &= \frac{\mathrm{R}^T\mathrm{R}}{\sigma^2} + \Sigma_{uu}^{-1} \Rightarrow {\Sigma^*}_q = (\frac{\mathrm{R}^T\mathrm{R}}{\sigma^2} + \Sigma_{uu}^{-1})^{-1} \\
\Sigma_q^{*-1}\mu_q^* &= \frac{\mathrm{R}^Ty}{\sigma^2} \Rightarrow \mu_q^* = \frac{{\Sigma_q}^*\mathrm{R}^Ty}{\sigma^2}
\end{aligned} \tag{2.84}$$

where, $\mathrm{R} = \Sigma_{fu}\Sigma_{uu}^{-1}$. Equation (2.84) provides the optimal variational distribution $q^*(u) \sim \mathcal{N}(\mu_q^*, {\Sigma_q}^*) \approx \mathcal{N}(\sigma^{-2}{\Sigma_q}^*\mathrm{R}^Ty, (\sigma^{-2}\mathrm{R}^T\mathrm{R} + \Sigma_{uu}^{-1})^{-1})$. Once the optimal variational distribution is estimated, it can be used to evaluate the posterior predictive distribution at test locations as

$$p(f_*|y) = \iint p(f_*, f, u)\,df du \tag{2.85}$$

Marginalizing the latent random variable f,

$$p(f_*|y) = \int p(f_*|u)\,q(u)du \tag{2.86}$$

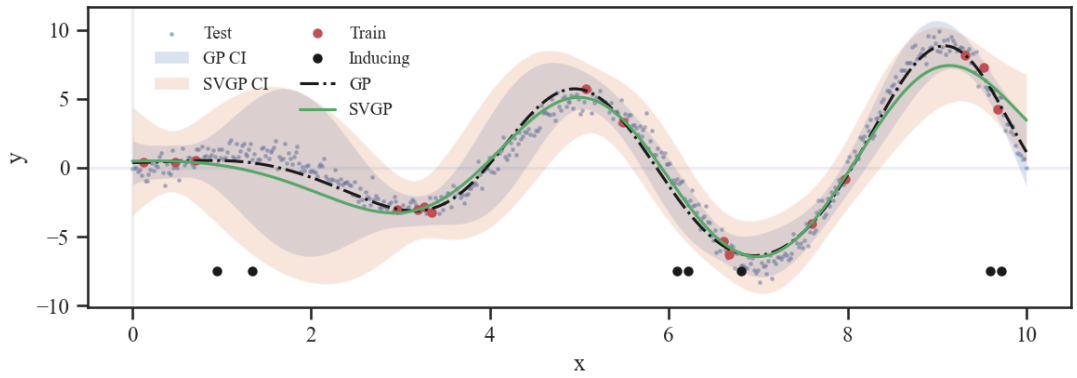
from the sparse prior given in equation (2.68), computing the conditional distribution of $p(f_*|u)$

$$p(f_*|u) \sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}u,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf_*}) \tag{2.87}$$

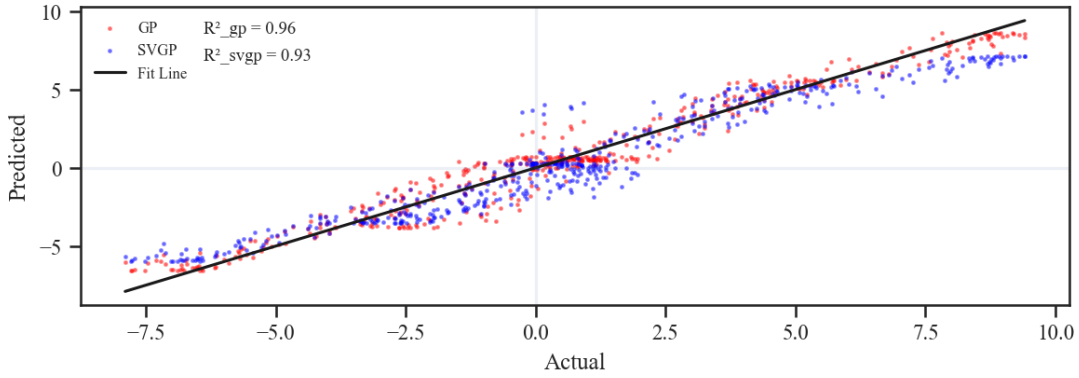
Finally, applying the Gaussian linear transformation rule, the posterior predictive distribution as a function of the variational parameters can be estimated as

$$\begin{aligned}
p(f_*|u) &\sim \mathcal{N}(\Sigma_{f_*u}\Sigma_{uu}^{-1}\mu_q^*,\ \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf_*} + \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_q^*\Sigma_{uu}^{-1}\Sigma_{uf_*}) \\
&\sim \mathcal{N}(f_*; \mu_f, \Sigma_f)
\end{aligned} \tag{2.88}$$

where $\mu_f = \Sigma_{f_*u}\Sigma_{uu}^{-1}\mu_q^*$ and $\Sigma_f = \Sigma_{f_*f_*} - \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_{uf_*} + \Sigma_{f_*u}\Sigma_{uu}^{-1}\Sigma_q^*\Sigma_{uu}^{-1}\Sigma_{f_*u}$. From equation (2.88), the mean trajectory and the associated 95% upper-lower confidence bounds are given by $y_{mean} = \mu_f$, $y_{upper} = \mu_f + 1.96 * \Sigma_f^{\frac{1}{2}}$ and $y_{lower} = \mu_f - 1.96 * \Sigma_f^{\frac{1}{2}}$ respectively.



(a) GP Vs SVGP predictive distribution

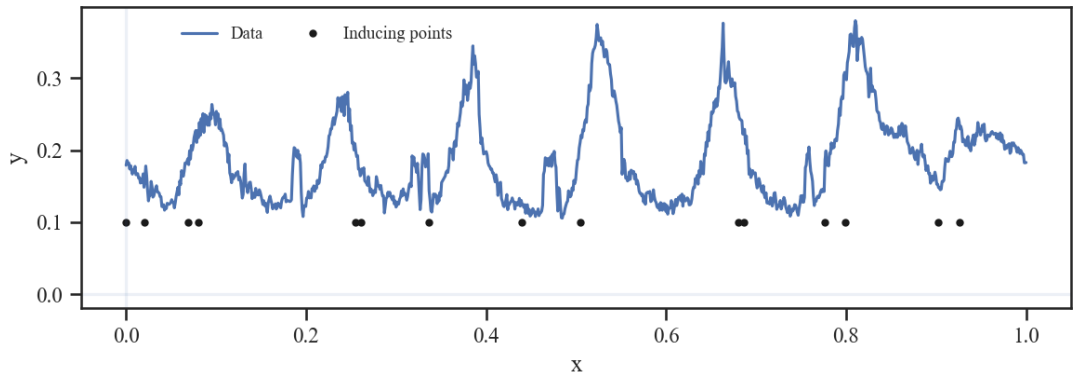


(b) R^2 model fit performance metrics

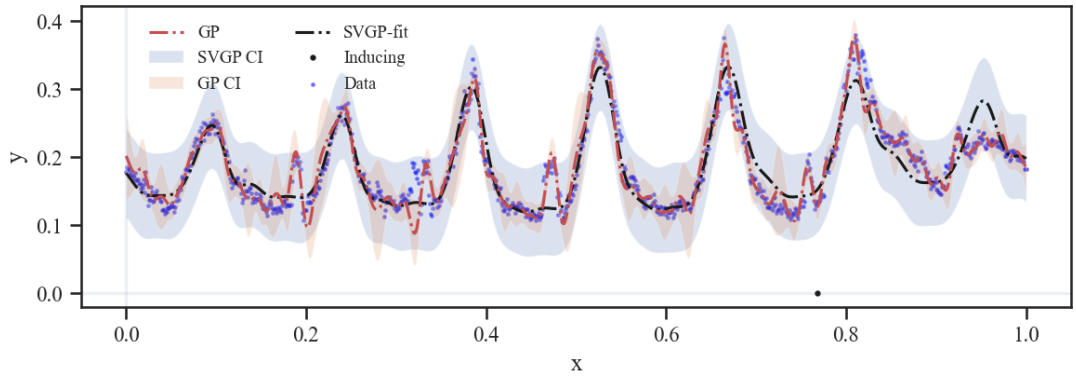
Figure 2.20: GP model approximation

The implementation of the Gaussian process in this manner brings two things into fruition. First, the computational burden will reduce from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ and the memory requirement from $\mathcal{O}(n^2)$ to $\mathcal{O}(m^2)$ where n and m are the number of data points and inducing locations respectively. Thereby ensuring the possibility of applying the Gaussian process to large datasets. Second, the the maximisation of the evidence lower bound only draws the variational posterior closer to the real posterior. Meaning the ELBO will always be lower than the marginal likelihood of the data. Hence, the ELBO serves as a testament to the fact that the resulting model will not overfit compared to the exact Gaussian fit with all the training data as shown in Figure 2.20. The Figure 2.20 shows the posterior distribution for an exact and sparse variational Gaussian models that are trained on samples drawn from a process $y = xsin(0.5\pi x), \ \ for\, x\, \epsilon\, [0, 10]$. A SE kernel for the covariance matrix, 7 inducing variables for the variational inference and 16 randomly selected points for the exact Gaussian were selected for model fitting. The variational approach predicts with less variance as compared to the exact Gaussian. However, the confidence interval, although broader in Figure 2.20, is generally dependent on the number of the inducing variables. A prediction with better accuracy and high confidence (i.e narrow confidence interval) can be achieved by increasing the number of inducing variables and optimising their location. However, for the computational efficiency, their number should be kept small compared to the number of training points. Though, the minimum threshold on the number of inducing variables required to represent a given distribution

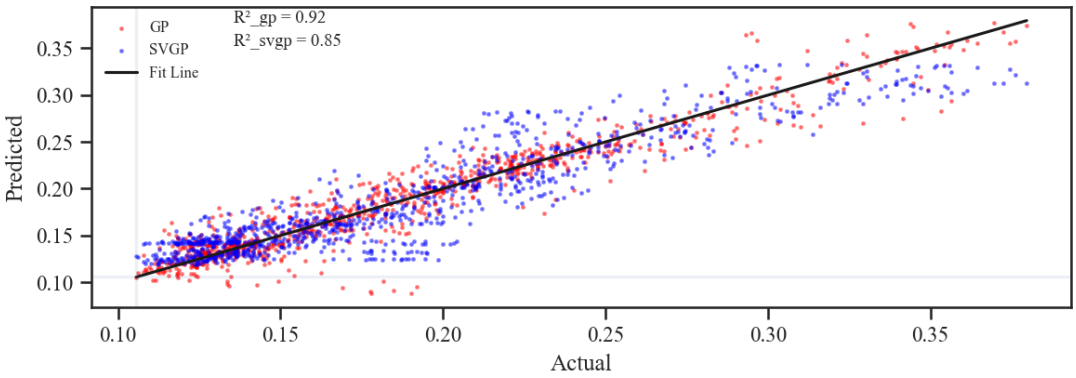
sufficiently is determined by the nature of data and the computational resources that one is willing to spare. Model approximation employing inducing variables lower than the required threshold returns a low order model approximation with smoother functions and high variance.



(a) A normalized weekly electricity consumption



(b) GP and SVGP model predictive distribution



(c) GP and SVGP R^2 model-fitness performance metrics

Figure 2.21: GP and SVGP predictive distribution for weekly power consumption

The size of the inducing variable is a design parameter. As such, tuning its value based on the desired performance criteria is not difficult. The challenging part of working with the sparse variational Gaussian models is designing the kernel matrix. The main task of the sparse approximation is to generalize the observed data with few selected points. And, generalization entails some aspects of the original data will be lost. It is an unavoidable outcome of using the model. However, by designing a suitable kernel combinations, the impact can be minimized. What type of basis kernels to select and how to combine them requires a bit of computation, patience and expertise. Real data is chaotic. Even in the presence of discernible patterns and prior expertise, determining the

nature of combination (i.e whether it is additive or multiplicative) and its overall impact on the predictive performance of the model needs continuous retraining. Consequently, it can get tiresome. For instance, Figure 2.21a, Figure 2.21b and Figure 2.21c show a slice of a normalized weekly electricity consumption data, GP and SVGP models predictive distribution along with R^2 performance metrics respectively.

The data was recorded at 10 minute interval with a total of 1008 observations per a week. It is a small data. However, the presence of multiple frequencies makes the kernel selection challenging. After multiple retraining, a kernel structure $k = k_1 + k_2 * K_3$ is selected for pattern discovery, where k_1 is SE kernel and k_2, k_3 are periodic kernels. For variational inference 16 inducing points and 160 randomly selected points for the Gaussian process fits provided a decent approximation to the original data. However, the predictive accuracy of the variational approximation can be improved further through an optimal combination of kernels. Figure 2.21 already shows computational efficiency of the SVGP model with a $\mathcal{O}(16^3)$ time complexity. This makes the algorithmic estimation of optimal kernel combination feasible and its exploration simplistic compared to the trailer and error search approach to suitable kernel functions.

2.7 Conclusion

In this chapter we have tried to review some of the non-parametric approaches to model building for a stochastic process using the Gaussian process. We have also seen its computational limitations along with a possible model approximations that would rectify it. The covariance matrix that gave the Gaussian process a higher degree flexibility in providing a fit function for a given problem, is also responsible for the computational limitations of the underlying model. The model computational need grows with the number of data points. The high memory storage, the cubic time complexity $\mathcal{O}(n^3)$ for matrix inversion during parameter optimisation and prediction makes the model undesirable in big data domain. This assertion is inspite of the computational advantage achieved through matrix decomposition methods such as the cholesky. But, such matrix factorization methods takes us only far enough when dealing with a millions of data points. Consequently, for the scalability and deployment of Gaussian models to large datasets, the number of points need to be minimized. This has been achieved through the model and posterior based approximations methods, such as the DTC, FITC and the SVGP.

The sparse representation using pseudo points have made it possible to condense the information content of a large data set. This has effectively minimized the dimension of the covariance matrix. As a result, the behaviour of the model and that of the data is no longer dictated by the correlation of all observed points, but rather few sampled points that can potentially summarize it. Consequently, the original data generation model, in the sense of the exact GP, is approximated by a model with a lower rank covariance matrix. As a result, some aspect of the original model and the possibility of exact inference is striped away for a gain in computational efficiency. This has been the reality in methods such as the DTC and FITC. However, techniques such as the SVGP have provided an alternative approximation methods in an effort to preserve at least the semblance of the original model.

Circumventing the Bayesian inference and embracing the path of variational approximation, the SVGP has rectified issues related to posterior intractability. Regardless of the specific methodology, approximation based performance improvement comes with the risk of a reduced variance estimation and predictive accuracy. The reduced predictive accuracy can be attributed to the smaller pseudo point approximation. The tendency to underestimate the variance however, depends on the specific modeling approach followed.

In the case of DTC and FITC, the conditional independence assumption between the latent and inducing variables plays the major part. On the other hand, in variational based approximations, the nature of the surrogate distribution that is used for the approximation tends to lower the estimated variance. For instance, attempting to approximate a complex data that follows an asymmetric distribution with a Gaussian ends up underestimating the variance.

In spite of that, the SVGP model has demonstrated some important qualities as a worthy predictive model for demand supply forecasting. Qualities such as, sparsity for computational efficiency; posterior-based approximation that guarantee the convenience of working on the original model; and the parameter learning using ELBO which puts an extra regularization so that the model doesn't overfit are some notable mentions. Furthermore, in addition to truly scaling the Gaussian process to big data, it has also made algorithmic kernel search techniques feasible. However, the suggested model approximation only rectify the computational impediments for the implementation of GP in big data domain. The predictive accuracy of the model is still dependent on suitability of the selected kernel regardless of the nature of the approximation. To that end, in the following chapter we will focus on search algorithms for optimal kernel evaluation using the SVGP as the underlying GP model.

Chapter 3

Kernel Estimation

3.1 Introduction

In the general machine learning framework, the availability of huge data offers ample opportunity to learn and infer educated generalizations. Unfortunately, the computational cost and efficiency of the GP model is tied to the size of the data [105]. As such, big data has been their Achilles heel and impeded their wider application. In chapter 2, we have discussed possible approximation methods that enhanced its efficiency and ensured its scalability. The proposed methods only mitigate issues related to speed of computation and memory utilization. They have nothing to do with improving the predictive accuracy of the model. In fact, the act of approximation by default degrades accurate data representation to a certain degree.

As a kernel-based learning model, the characteristics GP is primarily dependent on the kernel functions. By extension, the suitability of the kernels determine the generalization, inferential and forecast accuracy of the model. As such, the choice of kernel is an important design consideration during model development [105]. And yet, the question of how to select it appropriately in parallel with the problem at hand is not always clear. For instance, the dynamics of the model, the nature of the data and the desired outcome are some of the parameters that guide kernel selection. As such, choosing kernel requires prior expertise on the given problem, the observed pattern in the data, the future expectations, so and so forth. Especially, in a dynamic time series data with a high tendency to change, prior expertise might not be enough in selecting optimal kernels. Consequently, their selection mostly has relied on few well thought basis functions with an added trial and error approach.

Inherently kernel selection is computationally intensive. This has been one of the challenges in automatic optimal kernel evaluation. In spite of that, various methods have been suggested for algorithmic kernel estimation. For example, exhaustive, grid, randomized and nonparametric search methods are few notable mentions. However, the effectiveness of these approaches is dependent on the intricacies of the data and the frameworks in which they operate. For instance, for Gaussian models the dimension of the covariance matrix and continuous retraining presents a challenge for suitable kernel assessment. In the case of SVGP and MCMC-based models, the time complexity required for the ELBO and posterior convergence hinders the implementation of optimal search. Consequently, in addition to the respective search strategy, a computationally efficient kernel exploration should take into account the limitations of the underlying model.

In chapter 2, we have seen some of the kernel utilized for building an interpretable Gaussian model. In this chapter, we will investigate the implementation of SVGP for

electricity load forecasting, its limitation and the impact of kernel on predictive accuracy. Furthermore, to enhance the performance of the SVGP model, we will propose a stochastic approach to optimal kernel estimation and a GP approximation model based on random column sampling for a feasible kernel exploration.

3.2 Electricity demand profile

In this thesis, we employed both synthetic and real datasets. For a robust analysis some of the selected data follow a symmetric distribution and the rest are asymmetric. For instance, for the electricity load forecasting, the METRON energy company has provided us a two year total electricity consumption of more than fifty organizations. The collected demand profile is a time series data recorded for a period of two years at a ten minute interval. That means, we have a total observations of 144 and 1080 per a day and a week respectively. The demand profile exhibits a daily, weekly and yearly seasonality with an increasing trend as shown in Figure 3.1. However, a close inspection of the data shows there is a high irregularity in electricity demand due to the presence of Holiday's and dynamism of consumption. This has made the selection of suitable kernels challenging for the GP model development. Furthermore, the dependency of the models predictive accuracy on the appropriateness of the selected kernels makes the choice of kernels critical. A random combination of kernel using the recovered patterns and relying on the trial and error approach to multiple kernel learning doesn't provide the desired predictive accuracy. As such, in this chapter, taking advantage of the computational efficiency of the SVGP model, we will investigate optimal kernel search methods and their implementation in SVGP model for electricity load forecasting.

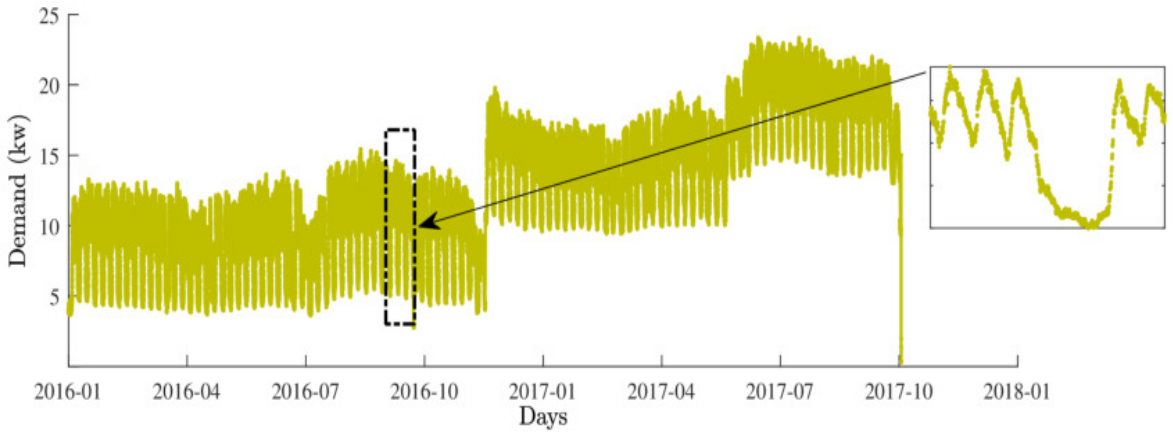


Figure 3.1: A two year total electricity demand for more than 50 organizations, collected at 10 minute interval(i.e 144 measurements in a day)

3.3 Optimal kernel combination

In principle the search for optimal kernel combination is similar to training and comparing an ensembles of Gaussian models trained on different covariance matrices. As such, the selection of optimal kernel combination falls to model evaluation based on a predefined performance metrics. There are a numbers of metrics and criterion's that one could apply to objectively compare these models. The average deviation of model outputs from the real value is one such quality metrics. As such, the mean squared error (MSE) and root mean squared error (RMSE) are the two popular approaches that are usually deployed in

evaluating model performance. However, for probabilistic models, the likelihood criterion is usually preferred. Regardless of the quality metrics employed, during model evaluation a kernel with the highest performance index will be the selected as the more suitable kernel for the problem at hand. In chapter 2, we have already stated unlike the parametric models, Gaussian process retain the training data during parameter optimisation and prediction. As a result, training a number of models for optimal kernel evaluation risks huge computational and memory storage requirements, especially for large data. However, the sparse and variational inference has rectified those computational bottlenecks and has made the Gaussian framework viable for automatic kernel evaluation.

3.3.1 Exhaustive kernel search

The Exhaustive kernel search is a brute force approach to optimal kernel evaluation. Given a set $\{k_1, k_2, \cdot\cdot, k_n\}\ \epsilon$ Q containing valid kernel functions, the exhaustive search approach finds all possible combinations. One positive contribution of this approach is that it guarantee and provides the right solution without leaving other unexplored alternatives. The need to evaluate every combination for suitability makes the technique computationally intensive. However, those risks can be managed by minimizing the number of kernel functions. Although inefficient, in the absence of clear insight on the structure of the data, this method can give an explanation on possible combinations better than the trial and error approach. The electricity consumption data shown in Figure 3.1 shows a linear trend, cyclic and periodic oscillations. This pattern discovery help in keeping the set of possible kernel function minimum. That is the very reason why we considered this approach for optimal kernel evaluation, despite its computational ineffectiveness.

Generally given a list of n kernels $\{$*Constant, SE, Periodic, Linear, RQ, Matern,*$\cdots\}\ \epsilon$ Q, where Q is the set containing all valid kernel functions, the exhaustive search construct a sub-list of kernel combinations in the order of

$$^{n}C_m = \frac{n!}{m!(n-m)!} \text{ where } m=1,2,\cdots,n+1 \tag{3.1}$$

In section 2.4.4, we have discussed that SE kernel can be used to model local similarities as well as global trends. Its combination with other kernels can model the dynamic degradation exhibited in a time series data. As such, it can be used as a base node to construct other possible mixtures. This assignment will minimize the number of the resulting mixtures. Every list represents a possible combination path from the base node to the child node as shown in Figure 3.14. As we go down from the base node (k_1) to the children ($k_2, k_3, ..., k_n$), the product rule of kernel construction given in proposition 2.4.2 is applied to combine the kernels. To test the validity of the resulting mixtures, a number of models based on the spares variational approximation given section 2.6.4 were trained to fit the observed data for each mixtures. The performance of each model is evaluated based on the mean squared error criterion and the optimality of their respective kernel is evaluated considering the width of the confidence interval and its coverage probability. These criterion were selected according to their relevance to the energy sector.

3.3.2 Evaluation metrics

Decision making in energy sector depends heavily on the confidence interval than the predicted mean trajectory. Accordingly, the mean prediction interval width (MPIW) and prediction interval coverage probability (PICP) are selected as a quantitative metrics to asses model predictive performance. These two measure the span of the interval and the percentage of forecast points falling within the interval respectively. By definition, the

width of the prediction interval is quantified through MPIW [20] and mathematically is given by

$$\text{MPIW} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i^{upper} - \hat{y}_i^{lower}) \tag{3.2}$$

where N is number of observations and $\hat{y}_i^{upper}$, $\hat{y}_i^{lower}$ are the upper and lower bounds of confidence interval for the i^{th} sample respectively.

On the other hand, PICP estimates the percentage of points falling within the confidence interval. To that end, we considered a binary variable $\zeta_i \, \epsilon \, \zeta$ that takes a membership value of 1 and 0; signifying the presence or absence of the point within the interval respectively. As such, PICP and the random variable ζ_i are mathematically defined as

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N}\zeta_i, \;\; where \;\; \zeta_i = \begin{bmatrix} 1, & \hat{y}_i^{lower} \leq y_i \leq \hat{y}_i^{upper} \\ 0, & otherwise \end{bmatrix} \tag{3.3}$$

The value of PICP falls between 0 and 1. A naive model with infinite confidence interval width can have a PICP value of 1. Consequently, a PICP value closer to 1 is considered a good quality indicator for a given model only when it is supported by a minimum MPIW value.

3.3.3 Model evaluation and kernel selection

The average mean squared error (MSE) between the predicted mean and the true value, is employed as a criterion for selecting the first z mixtures with the least MSE error as the best performing kernels. These kernels are again combined following the sum rule given in proposition 2.4.2 to form the final kernel which is given by

$$k = \sum_{i=a}^{z}(k_i) \tag{3.4}$$

Such manner of composing kernels compounds the number of hyperparameters and reduce efficiency. Therefore, the exact number of mixtures for the final kernel is left as a design parameter for the user to select based on the available computational resources. The

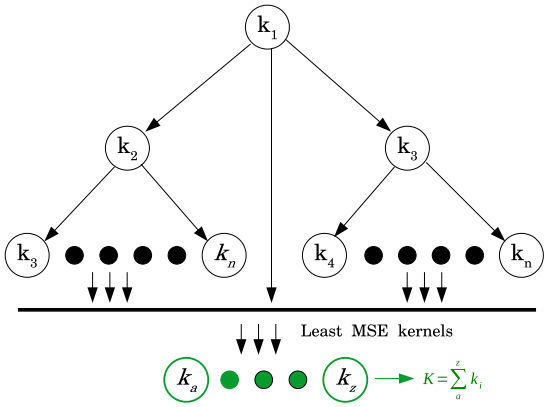


Figure 3.2: A combination of n kernels. z mixtures with the least mean squared error are selected to evaluate the final kernel $k = \sum_{i=a}^{z}(k_i)$ for model training and evaluation

algorithm is tested on a subset of the time series data given in Figure 3.1. In order to minimize the number of combinations, the SE kernel is taken out from the kernel definition and introduced via a multiplication as shown in Algorithm 1.

A number of models were trained for the demand profile provided in Figure 3.1. Since the data show a clear seasonality, we used a periodic kernel to model the daily, weekly and

Algorithm 1: Exhaustive kernel search

```
kernels ← {0 : k_1, 1 : k_2...n : k_n}
n ← kernels size
m ← kernels key list
k ← SE kernel
for index = 1, 2, ··, n + 1 do
    for list, in combination (m,index) do
        L ← list size
        for i= 0,1,. . . ,L do
            k ← k ∗ kernels[list[i]]
        end
        Train model using kernel k
        Estimate model output Y_mean
        Compute model MSE (Y_true − Y_mean)
        Collect model MSE error for each kernels
        k ← SE
    end
end
Evaluate the first z kernels with the least MSE error
return k ← ∑(zś kernels)
```

yearly seasonality along with other relevant kernels to capture the trend. A sliding window technique with a stride length $w_s = 1008$, and $w_s = 2016$ (for a week and two weeks ahead forecast respectively) is used for updating and forecasting. For the variational inference computation and ELBO optimisation, the number of inducing variables and maximum iteration are set to $z = 75$, and $Iteration = 14000$, respectively. We used the six best performing kernels to construct the optimal kernel. These kernel mixtures with their associated MSE scores are shown in Figure 3.3. To enhance accurate representation of the data, these composites are again combined to give the final kernel k

$$k = k_1k_4 + k_1k_4(k_2 + k_3) + k_1k_4k_2(k_3 + k_5) + \prod_{i=1}^{5} k_i \tag{3.5}$$

where k_1=SE, k_2= Matern, k_3= Periodic(daily), k_4= Periodic(weekly), k_5= Linear.

The SVGP model trained on the final kernel (k) given in equation (3.5), scored the lowest MSE relative to the other mixtures as shown in Figure 3.3. However, this kernel is not constant. It is one optimal combination based on the portion of the data and the hyperparameter setting. In a sense, the algorithm finds a suitable kernel dynamically, whenever the data or the hyperparameter settings change. We could minimize model complexity and further improve computational efficiency by reducing the number of mixtures in the final evaluation. A sample of the two weeks ahead demand forecast using the evaluated kernel is shown in Figure 3.4. The simulation showed the SVGP model's superior performance in handling big data. However, the sparse approximation degrades prediction accuracy for special days (i.e holidays) as shown in Figure 3.5. Generally, the model performs well for a regular day. But, despite the yearly periodicity introduced into the kernel to account for a holiday recurrence, the model fails to capture the yearly pattern. As a result, its fails to predict accurately the electric demand for holiday's. This is due to the uneven data distribution between the holidays and regular days. The bulk of the time series data is dominated by samples from regular (non-holiday) days. Therefore, the probability of the inducing variables distribution resembling this range is higher. Consequently, the model generalizes and considers holiday's just like any other day as

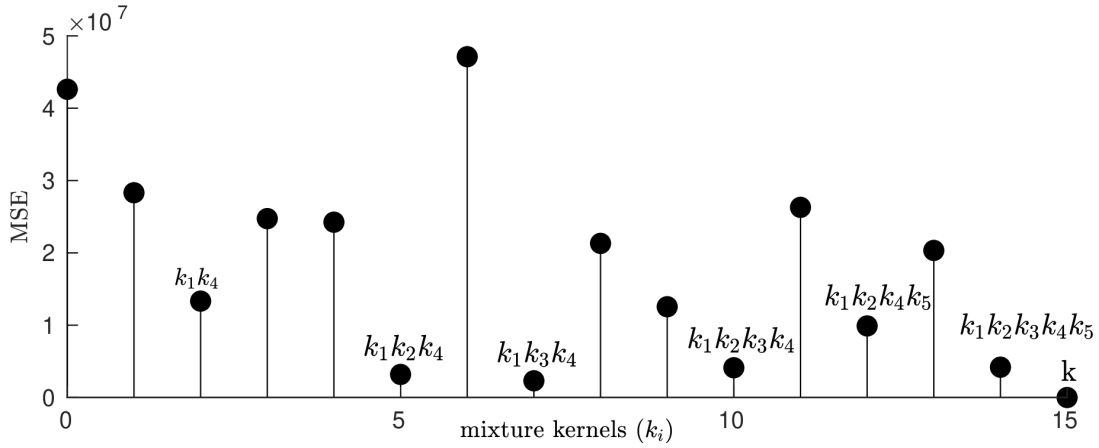


Figure 3.3: MSE score for the 15 kernel combinations. The final kernel is the sum of the six least scoring kernels whose MSE score displayed at index 15

shown in Figure 3.5

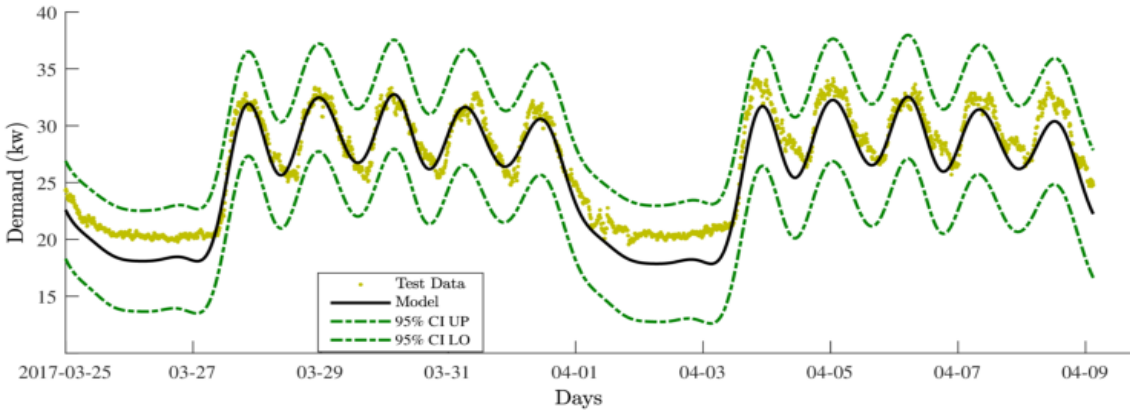


Figure 3.4: A two weeks ahead forecast for a predictive model with a year long prior observation and 75 inducing variables used for training and approximation. A window size = 1008

The inference drawn from few inducing variables and the associated prediction coverage is also dependent on the size of the training sample. This dependency is illustrated in Figure 3.6. Here, the model is trained with 4000 and 8000 samples using the same number of inducing variables. Increasing the sample size improves PICP by enlarging the MPIW. The reason being, expanding the sample size while holding the inducing variable constant introduces a higher uncertainty which inflates the prediction width. On the other hand, training the model on a smaller sample size resulted in a smaller MPIW which is followed by a reduction in the percentage of points covered under the prediction interval (smaller PICP). The simulations show large training sample is not a guarantee for better prediction. For example, in Figure 3.4, the model was trained on a year long prior observations using 75 inducing variables. A smaller MPIW and similar PICP can be achieved by training the model with 8000 samples as shown in Figure 3.6 and Table 3.1. Therefore, one can manipulate the sample size in conjunction with the number of inducing variable for a desirable prediction coverage and computational efficiency. In addition to sample size, PICP and MPIW are also affected by the size of inducing variables and their initial assignment. The experiment show their arbitrary placement and number assignment degrades prediction accuracy.

We later compared the performance of the model using the long short term memory

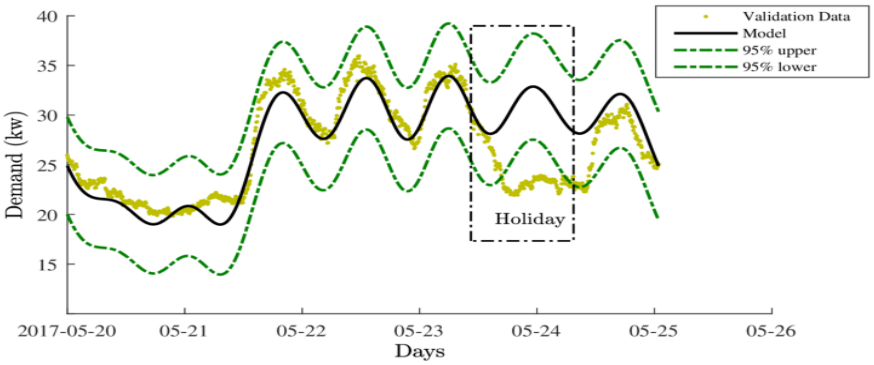


Figure 3.5: A week ahead prediction during holiday with 8000 prior observation and 75 inducing variables used for training and approximation

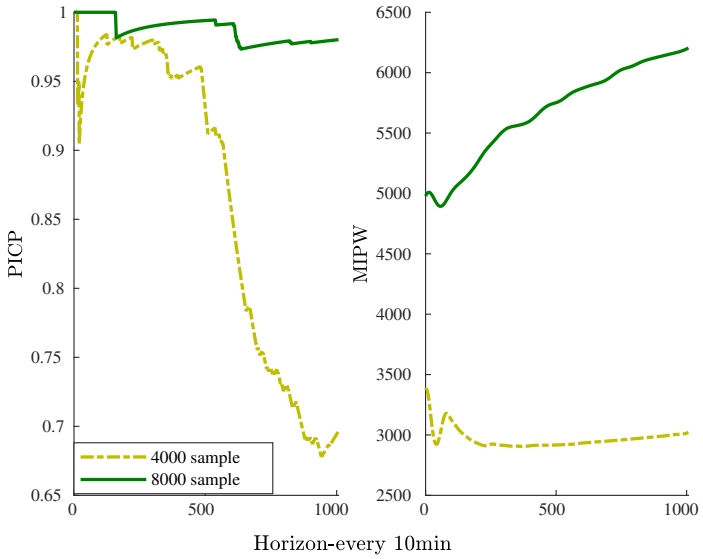


Figure 3.6: PICP and MPIW week ahead prediction for a model with 8000 and 4000 prior observation as training data

(LSTM) as a benchmark. LSTM neural nets provides point estimates only. They represent the latest non-parametric regression models that is efficient and score a fast computational time during forecasting. However, they do require a long training time and generate a trajectory for mean prediction only. In order equalize the comparison, we generated a forecast distribution through aggregating multiple LSTM models. Twenty one LSTM models were trained using the walk forward technique to generate a two weeks ahead prediction for the time series data shown in Figure 3.1. Each models consists of five inputs, an LSTM layer with 75 neurons followed by a dense layer and one output. We selected the number of neurons parallel to the number of inducing variables used when building the spares variational Gaussian model. In order to create a probabilistic irregularity on the forecast, we varied the batch-size, learning rate and epochs. Since we wanted to use the outputs of the 75 neurons, we didn't utilize drop out layer. For fast computation, the given data was re-sampled on hourly basis. The daily and weekly periodicity's are encoded and presented as an input feature along with the actual values. Then, the models were trained using 8400 samples on the training set. Finally, we made a two weeks ahead multiple-step forecast. The mean forecast and aggregated distribution are shown in Figure 3.7.

In regards to prediction latency, there is no measurable performance difference between a trained LSTM and Gaussian model. They both provide fast prediction in a fraction of seconds. However, in Table 1, we can see that the LSTM model trained on 6000 samples

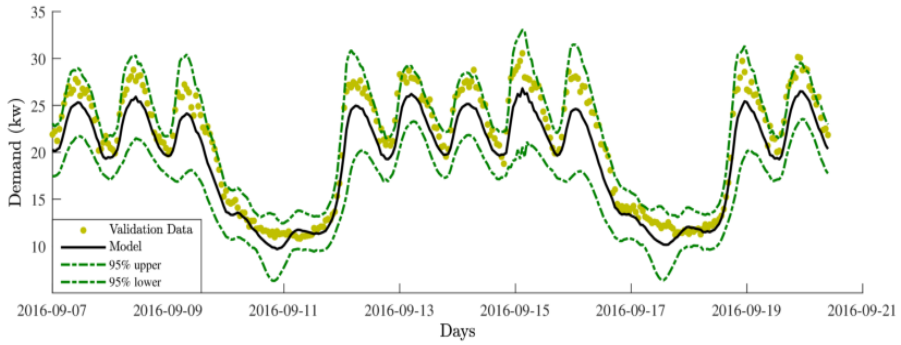


Figure 3.7: Aggregated lstm predictive model for a two weeks ahead demand forecast.

takes 244 seconds during training. By comparison, the Gaussian model takes 70 seconds in both cases. Aggregating LSTM models further expands the training time. In an online training and prediction where less latency is the objective, the Gaussian model outperform LSTM when it comes to fast training time and the convenience of providing a forecast distribution rather than a point estimate.

Table 3.1: Performance metrics

Model	Sample Size	PICP	MPIW	Training Time
LSTM	6000	0.935	6.58kw	244 sec
SVGP	8000	0.989	7.33kw	70 sec
	4000	0.962	7.83kw	70 sec

Kernel based learning through exhaustive kernel search method exposed few things. In addition to the inherent inefficiency of the search algorithm, the variational approximation requires a considerable amount of time for posterior convergence. The variational approach provides a tangible objective function that can be used as a metric to measure the degree of approximation during optimization. However, multiple model training and optimization requires a lot of computation and iteration which linearly increase the computational time required for optimal kernel evaluation. As such, any attempt on the algorithmic kernel evaluation requires a continuous training and evaluation of the model. This will ultimately affect the computational time and efficiency of the model. Consequently, in addition to the exploration mechanism, an effective implementation should address the limitation of the underlying model. As such, a faster and more practical approach in model building that is scalable to large data must be followed.

The sampling theorem states that few point have the capacity to summarize and capture the information content of a given data. The fidelity of this representation depends on many factors like the sampling rate, the range of frequencies so and so forth. However, it makes a valid argument that for pattern discovery, analysing the entire data is not necessary. Hence, the easiest alternative would be a Gaussian process model that is based on a randomised column sampling. In doing so, it results in a computationally feasible model to carry out the search algorithm. This sparse approximation makes the size of the kernel dimension manageable through sampling. The reduction in kernel size together with the Bayesian inference offers a faster model evaluation compared to other approximations. As such, this arrangement rectify the time constraint that would have been required for the ELBO convergence. The rationalization behind such model building and kernel estimation can be attributed to the fact that a kernel that approximate a sampled points could approximate the original data as well.

3.4 Random sparse gaussian approximation

A Gaussian model based on random sampling doesn't need the entire input space for training. It represents the given observation by a selected subsets of the data that could potentially generalize it. This approximation will transform the data space from $\mathcal{D}\ \epsilon\ \mathbb{R}^{NxN}$ to $\overline{\mathcal{D}}\ \epsilon\ \mathbb{R}^{pxp}$, where p is number of points selected for approximation. However, the accuracy of representation depends on how well the selected data $\bar{D}$ summarize the original data.

Hence, given a time series data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ and a set of randomly sampled points $\overline{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^{P}$, it defines a joint distribution over the sampled points f and all other points f_* as

$$p(f, f_*) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\overline{\Sigma}_{pp} & \Sigma_{np}\\ \Sigma_{np}^T & \Sigma_{nn}\end{bmatrix}\right) \tag{3.6}$$

where $\overline{\Sigma}_{pp}$ the approximating kernel, n is the size of training data and p is a randomly selected points without replacement. The principle of model fitting and parameter optimisation follows the regular Bayesian framework discussed in section 2.5.2. Here, the only difference being, the prior definition in equation (3.6) manipulate a kernel with a reduced dimension.

The sampled observations (y_p) are assumed to be samples of the random variable $y(x)$ which are drawn randomly from a normally distributed Gaussian with mean $f \approx 0$ and variance $\epsilon \sim \mathcal{N}(0, \beta^2 I_p)$,

$$y(x) \sim \mathcal{N}(0, \overline{\Sigma}_{pp} + \beta^2 I_p) \tag{3.7}$$

As such, the likelihood of the data $p(y|f)$ can be expressed as

$$= \frac{1}{(2\pi)^{\frac{n}{2}} det(\overline{\Sigma}_{pp} + \beta^2 I_p)^{\frac{1}{2}}} exp\left(-\frac{y(x)^T(\overline{\Sigma}_{pp}+\beta^2 I_p)^{-1} y(x)}{2}\right) \tag{3.8}$$

The parameters of the kernel matrix (i.e $\theta_i, i...m$) defining the covariance between the two random variables are estimated through

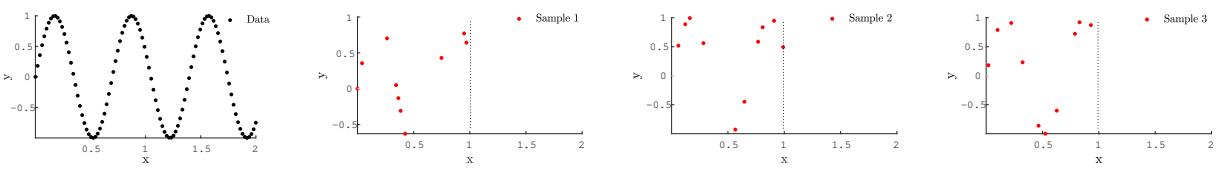
$$\frac{\partial log(p(y|f))}{\partial \theta_i} = 0 \tag{3.9}$$

Performing marginalization and conditioning on (3.6) - (3.8), the Bayesian framework defines the posterior distribution $p(f_*|f, y)$ as

$$p(f_*|y) = \mathcal{N}(f_*; \mu_f, \Sigma_f) \tag{3.10}$$

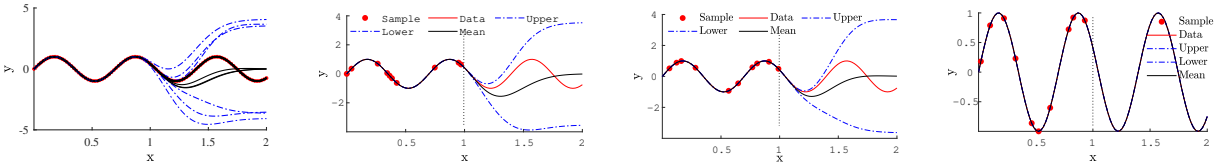
$$\sim \mathcal{N}\left(\left[\Sigma_{np}(\overline{\Sigma}_{pp} + \beta^2 I_p)^{-1} y, \Sigma_{nn} - \Sigma_{np}(\overline{\Sigma}_{pp} + \beta^2 I_p)^{-1}\Sigma_{pn}^T\right]\right) \tag{3.11}$$

where $\mu_f = \Sigma_{np}(\overline{\Sigma}_{pp} + \beta^2 I_p)^{-1} y$ and $\Sigma_f = \Sigma_{nn} - \Sigma_{np}(\overline{\Sigma}_{pp} + \beta^2 I_p)^{-1}\Sigma_{pn}^T$. From (3.11), the mean forecast and the associated 95% upper and lower prediction bounds are given by $y_{mean} = \mu_f$, $y_{upper} = \mu_f + 1.96 * {\Sigma_f}^{\frac{1}{2}}$ and $y_{lower} = \mu_f - 1.96 * {\Sigma_f}^{\frac{1}{2}}$ respectively.



(a) $y = sin(9x)$ (b) Random draw 1 (c) Random draw 2 (d) Random draw 3

Figure 3.8: Input space sampling



(a) Ensembles (b) Draw 1 forecast (c) Draw 2 forecast (d) Draw 3 forecast

Figure 3.9: Gaussian process predictive distribution on sampled space

The objective in the Gaussian process with random sampling is to select the right combinations of input columns and kernel parameters so that the original likelihood of the data $y(x) \sim \mathcal{N}(0, \Sigma_{nn} + \beta^2 I_n)$ can be approximated by $y(x) \sim \mathcal{N}(0, \overline{\Sigma}_{pp} + \beta^2 I_p)$, where $\overline{\Sigma}_{pp}$ is a low rank approximating kernel. Though, the mathematical representation looks sound, the validity of this approach is dependent on the sampled input columns in approximating the whole observation. As such, the question that remains is how to find and select those approximating points without the need for inducing variables or the trouble of optimising their location. One potential alternative is to sample the original data by applying the Nyquist-Shannon criteria. Especially, for large data, such technique can reduce the sample space for random draws. Consequently, it can provide a set of summarising points with high probability for accurate representation of the original observation. Accordingly, at the preprocessing stage of data preparation, the original data can be down-sample at frequency f_s to provide the sample space S

$$f_s = 2 * f_{max} \tag{3.12}$$

where f_s and f_{max} are the sampling frequency used and maximum frequency observed within the data respectively. Then, a training data $\{(x_i, y_i)\}_{i=1}^{P}$ is created iteratively by drawing P samples randomly without replacement from the sample space $\{(x_i, y_i)\}_{i=1}^{P}$ ϵ S. An m number of Gaussian models will be trained and evaluated by sampling the sample space S randomly without replacement. Consequently, it will create an ensembles of Gaussian models with a total time complexity $\mathcal{O}(m\mathrm{P}^3)$ where m and P are the size of iteration and the number of sampled points. This process is better contextualized and illustrated in Figure 3.8 and Figure 3.9. From the original data given in Figure 3.8a, the first half is used for sampling. The second half and the leftovers points from sampling are later used for testing and model validation. The first half is sampled randomly, ten points (i.e P= 10) at a time without replacements as shown in Figure 3.8b - Figure 3.8d. Then, three models are trained using the selected clusters and their predictive distribution is shown in Figure 3.9b - Figure 3.9d. In Figure 3.9, the best approximating model is the one in Figure 3.9d and it is chosen via a predefined performance metrics (i.e lowest MSE score, PICP, MPIW, etc) or on the basis of which model approximate the original data best.

3.4.1 Model evaluation via cross-validation

For the sake of the sparse approximation, we followed a sampling approach for training point selection. This has effectively reduced the size of data, a data which could have been utilized for improving the inferential ability of the model. In the presence of a number of Gaussian models to evaluate and less data to work with, the cross-validation approach is the best technique to apply for model evaluation. In reality, it is a scheme that is computationally intensive. However, it is also a valid approach that guarantee efficient utilization of the training data [37, 89, 104]. The sampling divides the training data into two. Here, there is a 2-fold cross-validation with two unequally sized subsets

of data. In this scheme the model is trained continuously on a new randomly generated P sampled points. The model is evaluated on the remaining P - 1 data points. The randomness in column sampling rectify the selection bias that may be introduced when preparing the data from training. For a holistic model comparison, the root mean square error (RMSE) and R^2 score are utilized as the performance metrics. As a result, the model and the training points will be accepted or rejected as a representative to the underlying generative model based on which sampled columns fulfilled the desired performance index. For instance, given a time series data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, 2-fold cross-validation, provides $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{p}$ as the training data and the model is evaluated on $\mathcal{D}_v = \{(x_i, y_i)\}_{i=p}^{n}$, so that the average RMSE on the validation set is given as

$$\text{RMSE} = \sqrt{\sum_{i=p}^{n} \frac{(y_i - \hat{y}_i)^2}{n - p}} \tag{3.13}$$

where y_i and $\hat{y}_i$ are the observed value and predicted value at point i. As a result, the model and the training points will be accepted or rejected as a representative to the underlying generative model based on which combinations fulfilled the desired index. The random data selection and the manner in which optimal model performance is accepted or rejected gives the notion of Monte Carlo in model building and evaluation as shown in Algorithm 2.

The proposed sparse model approximation perform a fast fourier transform (FFT) on the input data to retrieve the maximum frequency f_{max}. This frequency is used to estimate the required sampling frequency f_s. Points located at the sampling frequency contain enough representative information for the whole observation. As such, the given data is down-sampled to create subset of points for model training. Once the training domain is created, we randomly select P points iteratively for model fitting and parameter optimization. The sampling is carried out without replacement. Therefore, the remaining points will constitute the majority of the test data. The root mean squared error on the test data is applied as a criterion for model comparison. The model with the least mean squared error is finally taken as the best performing model as shown in the algorithm 2.

3.4.2 Random column sampling simulation

It is an established fact that all data contains a little bit of noise either systematic or random. For the FFT analysis, We assumed the maximum frequency as the highest frequency with a non-zero signal strength. This assumption creates a problem when analysing data with added noise. Especially, when the modeling approach followed is based on a handful of points that are thought to potentially summarize the underling data distribution. The frequency spectrum of a noise signal is vast. Determining the sampling period in the presence of a noise affects the acquisition of subset of data needed for training. Consequently, the noise must be filtered out before performing dimensional reduction of the given data. As such, data analysis and determination of a cut-off signal strength is necessary to nullify noise frequencies.

We tasted the random sparse Gaussian approximation (RSGA) algorithm on a synthetic data. We used a non-linear function that exhibit periodicity and gradual degradation to encompass features of a time series data

$$\begin{gathered} y = sin(2\pi f_0 * x) + sin(2\pi f_1 * x) + sin(2\pi f_2 * x) \\ with\ \epsilon \sim \mathcal{N}(0, \beta^2) \\ x\ [0, 100], f_o = 0.08, f_1 = f_0/10, f_2 = f_0/2 \end{gathered} \tag{3.14}$$

whose time and frequency spectrum are shown in Figure 3.10a, Figure 3.10b respectively. The data exhibit periodic oscillation and gradual decay. As such, the exponential squared

Algorithm 2: Random sparse gaussian approximation

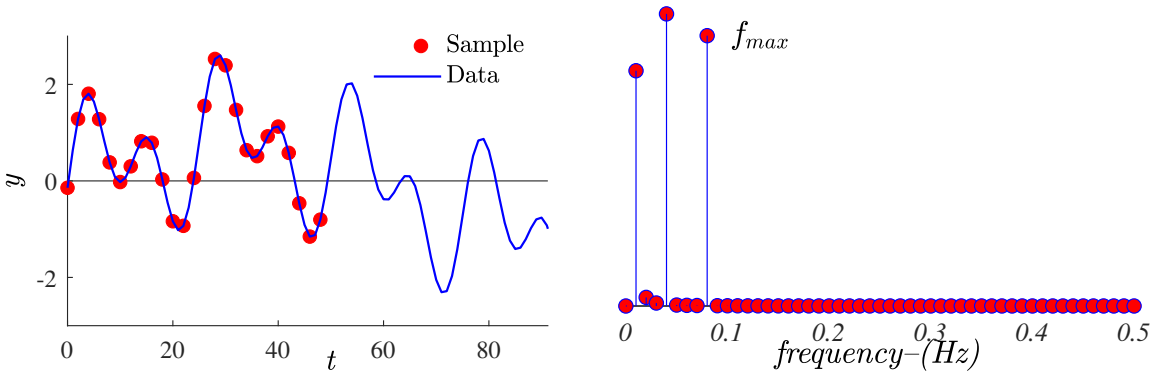
Input: data (X,Y)
Initialize kernel hyper-parameter θ = [σ^2,l,p,...]
Initialize P : number of points for fitting
Define kernel structure $\overline{\Sigma}_{pp}$
Determine maximum frequency $f_{max} = max(\text{FFT(Y)})$
Define sampling frequency $f_s = 2 * f_{max}$
X_s, Y_s = downSample(X,Y,f_s)
samples = [0,1,2,...,len(X_s)]
indices = random.choice(*samples*,p,replace=False)
X_p, Y_p = X_s[*indices*], Y_s[*indices*]
model = GP($\overline{\Sigma}_{pp}$)
θ_{opt} = model.fit(X_p,Y_p) and optimize parameters
Y_m = model.predict(X_s)
$score_{opt}$ = model.score(Y_m, Y_s)
X_{opt}, Y_{opt} = X_p, Y_p
for *index* = $1, 2, \ldots, m$ **do**
  indices = random.choice(*samples*,P,replace=False)
  X_p, Y_p = X_s[*indices*], Y_s[*indices*]
  θ = model.fit(X_p,Y_p) and optimize parameters
  Y_m = model.predict(X_s)
  score = model.score(Y_m, Y_s)
  if $i = 1$ **then**
    θ_{opt}, $score_{opt}$ = θ, *score*
    X_{opt}, Y_{opt} = X_p, Y_p
    else
      if *score* < $score_{opt}$ **then**
        θ_{opt}, $score_{opt}$ = θ, *score*
        X_{opt}, Y_{opt} = X_p, Y_p
      end
    end
  end
end

and periodic kernel functions are used used for modeling the local and global co-variances.

$$\overline{\Sigma_{ij}} = \sigma^2 exp\left(-\frac{||x_i - x_j||}{2l^2}\right) + \sigma_p^2 exp\left(-(\frac{2sin^2(\pi||x_i - x_j||/p)}{l_p^2}\right)$$

The random sparse Gaussian model was trained and evaluated on a 100 generated points. The first 50 points were selected for training. The training set was resampled again with $1/2f_{max}$ sampling period and further reduced to 25 points. Then iteratively 10 points were randomly selected without replacement for model fitting and the remaining 40 points were used for a continuous model evaluation. A multi-start minimization is followed for the kernel hyperparameter optimisation. As such, the optimiser restart was set to 50 to randomly sample the parameter space.

The experiment shows that random column sampling without replacement creates a better convergence as compared to sampling with replacement counterpart as shown in Figure 3.11b. The acceptance and rejection of a viable model is also demonstrated in Figure 3.11b. The algorithm will keep the same state (i.e model, parameters, sampled input locations) while rejecting all other models with a higher MSE score as shown in algorithm 2

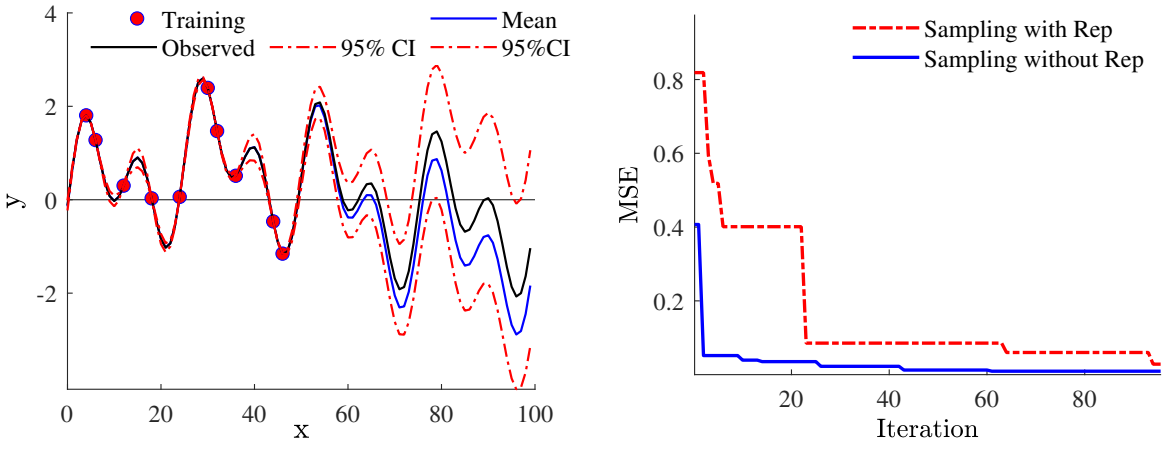


(a) Data with used for training (b) Data frequency spectrum

Figure 3.10: Data with its frequency spectrum

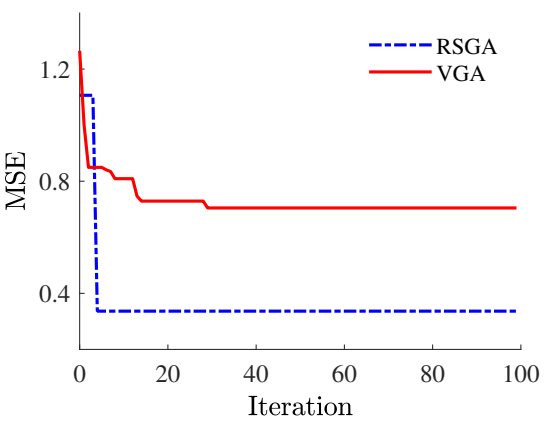
and the horizontal lines in Figure 3.11b. This algorithm offers by far the easiest, computationally efficient and scalable Gaussian model for algorithmic kernel search. Although the fourier based sampling helps in reducing the search space for potential summarising points, the sampling scheme is undirected. As such it exhibit a stochastic nature.

Later on, the RSGA algorithm is compared with the variational Gaussian approximation (VGA). We used the same kernel for data structure discovery. For uniform comparison, we set the inducing variables $z = 10$ and iteration to 20000 for variational inference and ELBO maximization. The experiment shows that the random Gaussian model (RSGA) exhibits faster convergence and a better predictive performance than the variational Gaussian (VGA). Compared to the variational, the randomize approximation scored a lower mean squared error (MSE) and a higher R^2 score as shown in Figure 3.12. In Figure 3.12a and Figure 3.12b, the MSE and R^2 scores prove that the RSGA approximate and explain the variations in the observed data better than VGA. Variational inference requires a higher number of iteration for better accuracy. This affects the computational efficiency of the model. Since it is not based on variational inference, RSGA exhibits a faster training time and under a good summarizing points, a higher predictive performance. Furthermore, the experiment shows modeling building and evaluation through a random column sampling is 10 times faster than the variational alternative. This can be attributed to the iterative computation required in variational inference for ELBO maximization. In this particular experiment, model building and evaluation took $t_e \leq 2.5$ seconds in RSGA while taking $t_e \leq 25$ seconds on average in case of VGA.

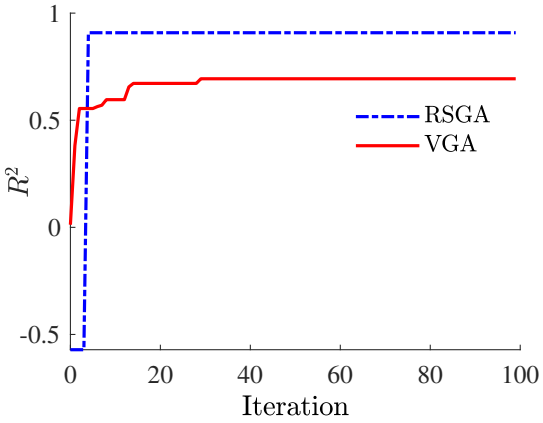


(a) A predictive distribution using RSGA (b) Effect of sampling with/out replacement

Figure 3.11: Predictive distribution with effect of sampling



(a) RSGA and VGA MSE convergence

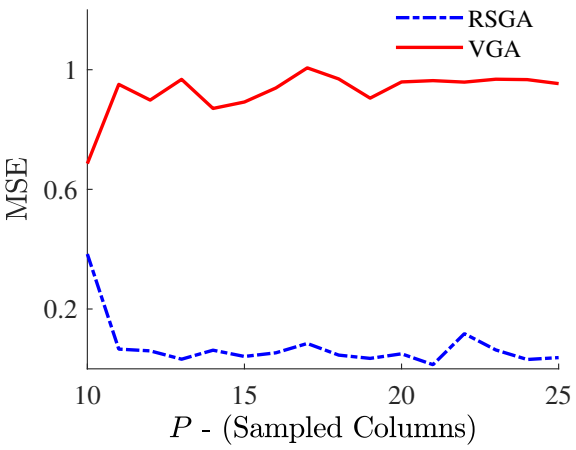


(b) RSGA and VGA R^2 model fitness score

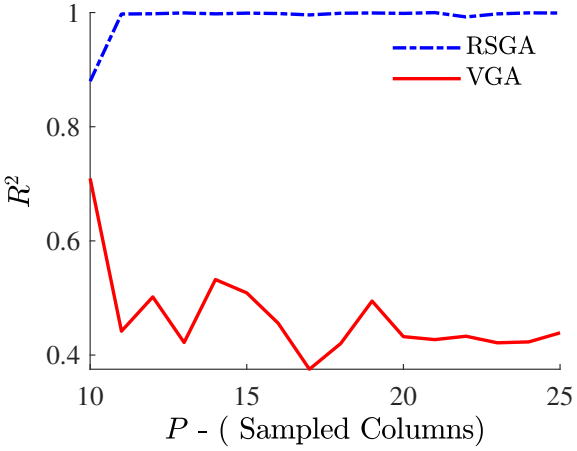
Figure 3.12: Random sparse (RSGA) Vs variational (VGA) gaussian approximation

The effect of sampled columns were one aspect of the model building process that we were interested in. As such, we carried out an experiment to determine how the number of sampled columns P affect the predictive performance. We varied the sampled columns $p \epsilon [10, 26]$ and iteratively trained more than 300 models based on the RSGA and VGA.

The experiment showed increasing the number of sampled columns improved the predictive performance of the RSGA as shown in Figure 3.13a and Figure 3.13b. However, the most striking difference in regards to the variational inference was the computational efficiency. The time required for model training in VGA is on average 10 times more than the RSGA. As such, the simulation proved that random column sampling is more computational efficient than the variational alternative. This approach could also be used



(a) RSGAVs VGA MSE score for different P



(b) R^2 score of RSGA Vs VGA for different P

Figure 3.13: Random Vs variational model comparison for sampled column Pϵ[10, 25]

to determine the minimum number of columns (i.e number of inducing variable and their location in VGA) that are required to approximate the given data using a predefined performance metrics such as R^2 or MSE. The RSGA model achieved the maximum but similar R^2 score for all column sampling $P \geq 12$ as shown in Figure 3.13b. As such, together with the MSE score in Figure 3.13a, we can say that $P = 12$ are the minimum number column samples that are required to achieve the needed performance index. Now that the approximating columns for kernel size reduction is done, in the next section we will review an efficient method for kernel exploration. We will see how to select appropriate kernels for similarity measure in a way that guarantee both sufficient data representation and computational efficiency using the random column sampling as the underlying model

building framework.

3.5 Stochastic kernel search

The stochastic kernel search uses the same compositional learning listed in proposition 2.4.2 to construct kernel from primitive basis functions for pattern discovery. These techniques were used in the exhaustive search methods. However, unlike the exhaustive search method, only promising kernels that could provide the highest data representation are added to the mix. The approach begins by checking for local similarity and then extends this to a global correlation. To improve the search efficiency, the primitive kernels are divided into groups for local and global similarity before the search begins. Among the available basis function, SE function, RQ function, and Matern basis functions are selected to model local similarity. The remaining kernel functions are grouped in a set for global similarity. Such arrangement will constrain the search space and rectify the time complexity incurred during the random selection and evaluation phase. Once the kernels are ground, the next step is to train the Gaussian model to select the best local kernel. To that end, the root mean squared error RMSE is used as a criterion for model evaluation. Once a suitable local kernel is established, it will be used as a based node to construct other possible mixtures following proposition 2.4.2. This procedure creates a

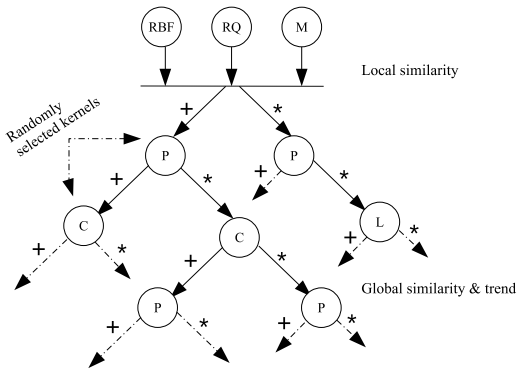


Figure 3.14: Kernel combinational tree using the product ($*$) and the sum ($+$) rule. Here P,L, and C stand for the periodic, linear and constant kernel respectively. Kernels on the path that results in the least RMSE score, are selected as the suitable mixtures for the given data

tree of possible kernel combinations as shown in Figure 3.14. The resulting mixtures are ranked by iteratively evaluating the random Gaussian model (RSGA) given in algorithm 2. The transition from the node to node is dictated by which path results in the minimum RMSE score. As long as the path offers the lowest error the branches are expanded up to a user defined depth parameter d. According to proposition 2.4.2, applying the sum and product rule, a single node bores two child nodes. If the RMSE score of the child nodes are greater than the parent node, the stochastic search will stop branching. Then, the kernel mixtures from the base node to the last parent node will be forwarded as the combinations that best explain the observed data. The pictorial description and the details of the search mechanism is outlined in Figure 3.14 and algorithm 3. Inherently compositional learning compounds the number of hyperparameters. As a result, a parameter d is provided to monitor the level of complexity and depth of branching.

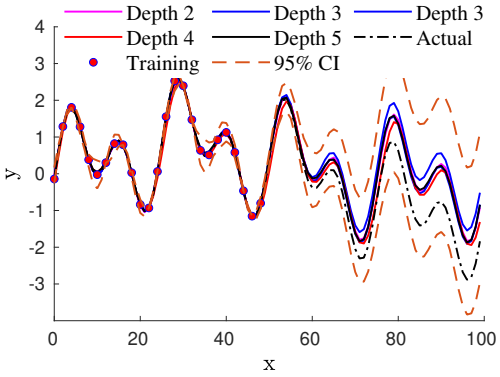
Algorithm 3: Stochastic kernel learning

```
Given kernel dictionaries for local(L) and global(G) Cov
L = {'se': SE, 'rq': RQ, 'm': Matern}
G = {'p': Periodic, 'c': Const, 'Wh': Noise,'L': Linear}
Given the depth = n for child nodes
optk = [ ]
score = [ ]
for key, k in L.items() do
    Evaluate GP for kernel k and return RMSE
    if score is empty then
        optk = k
        score = RMSE
        else
            if RMSE < score then
                optk = k
                score = RMSE
            end
        end
    end
end
for i in range of len(G) do
    Randomly select a kernel k from G
    Generate tk = optk + k and tk = optk * k
    Evaluate GP for each mixtures and return RMSE
    if RMSE < score then
        optk = tk
        score = RMSE
        break
        else
            continue
        end
    end
end
for i in range of (depth - 1) do
    Randomly select a kernel k from G
    Generate tk = optk + k and tk = optk * k
    Evaluate GP for each mixtures and return RMSE
    if RMSE > score then
        break
        else
            optk = tk
            score = RMSE
        end
    end
end
```

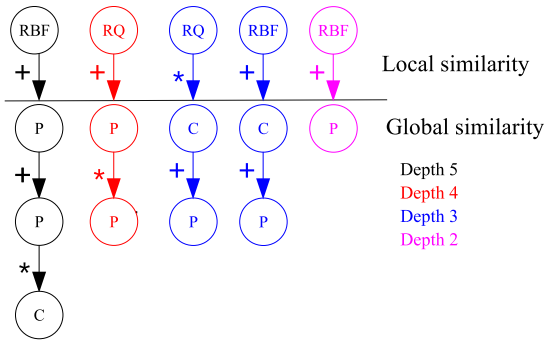
3.5.1 Stochastic kernel search simulation

The stochastic search algorithm was tested on a real and toy data for the purpose of generalization. For the synthetic data, we selected a non-linear function given in equation (3.14). The random sparse Gaussian model was trained on a 100 generated points. The first 50 points were used for training and evaluation. The training set is further reduced to 25 points by resampling it at $1/2f_{max}$. A stochastic data sampling was carried out for RSGA model training and evaluation as given in algorithm 2. The kernels were combined using the stochastic compositional kernel learning approach given in algorithm 3. For the iterative model training, 10 points were randomly selected without replacement. Then, the remaining 40 points were used for continuous model evaluation. For kernel hyperparameters optimisation, a multi-start minimization with a 50 optimiser restart was applied. For pattern learning the tree depth was varied from 2 to 5.

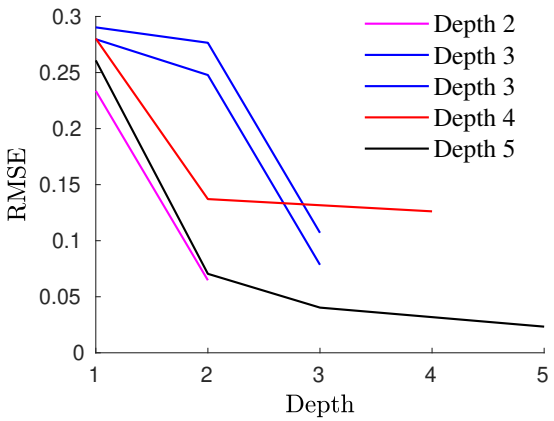
In the experiment, the algorithm combines elementary kernels stochastically. As such, in every iteration different compositional kernels were given as a potential mixtures outputs as shown in Figure 3.15b. The experiment was carried out with a maximum tree depth of 5. After the first node is fixed, using the provided kernels for local similarity, the algorithm uses the RMSE criterion to find additional nodes for global similarity. Traversing the tree along the least RMSE path allowed the algorithm to branch off stochastically and find other suitable mixtures as shown in Figure 3.15b and Figure 3.15c. The observed randomness can also be attributed to the stochasticity introduced during training point selection. Despite the depth, the estimated complex kernels delivered more or less similar predictive performance as compared to the simple composites as shown in Figure 3.15a and Figure 3.15b. The effect of compositional learning on the time complexity of the model



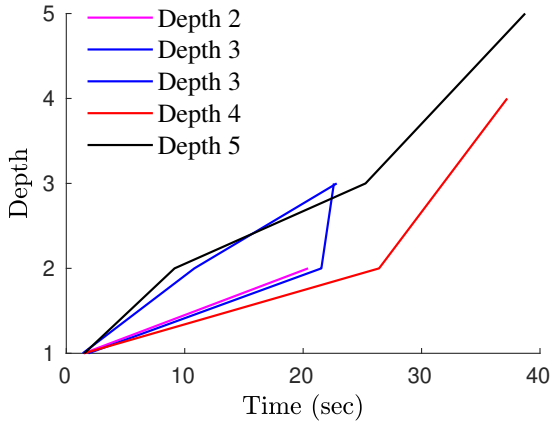
(a) Kernel with depth 2-5 prediction



(b) Stochastic compositional kernel learning



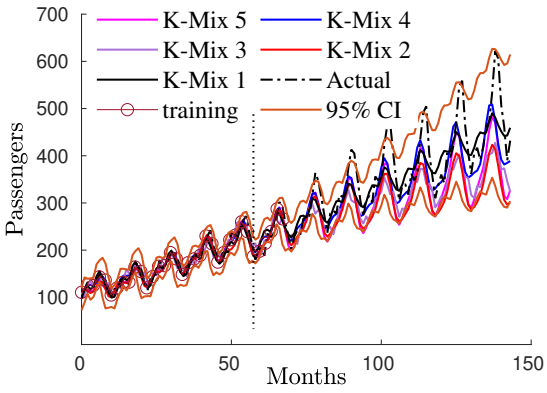
(c) RMSE for kernels with depth 2-5



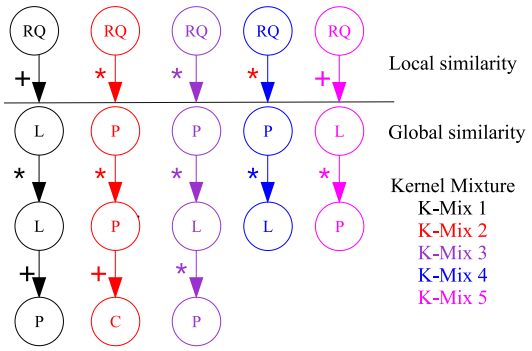
(d) Kernel learning time complexity

Figure 3.15: Model predictive distribution and compositional kernel learning

can be seen in Figure 3.15d. Figure 3.15d shows as more and more kernels are added



(a) Distribution for kernel-Mix 1 - 5



(b) Possible kernel mixture

Figure 3.16: Passengers flight data predictive model and possible kernel combinations

to the mix, the time required to optimise the hyperparameters of the model increase. Thereby increase the overall model evaluation time. The algorithm can uncover any hidden pattern provided enough tree depth. This in return increase the number of kernels in the mixture. Combining kernels in such a manner compounds the number of hyperparameters to optimize. Consequently, introducing an exponential time requirement for model estimation as shown in Figure 3.15d. As such, considering the available resources and the desired predictive performance, one could manipulate the depth for the exploration of an appropriate kernel. The experiment also showed, some of the returned mixtures were not able to accurately explain the data or provide the desired level of predictive performance. This is due the fact that the exploration and the formation of new branches is predicated upon scoring the least RMSE error. If both rules (sum and product) of combining kernels resulted in a higher RMSE error compared to the immediate parent node, the algorithm stops the exploration prematurely. Therefore, it is necessary to retrain and assess the returned kernels to determine if they accurately explain the training data.

Later on, the same strategy was applied to evaluate different kernel configurations for a passenger flight data which exhibits trend and seasonality. The data was divided using a train-test split of 50-50 for training and evaluation. We further down-sampled the training data and iteratively selected 10 points randomly without replacement for model training. The experiment showed, the presence of trend affects the time efficiency of the search and the optimality of all returned kernels. However, the algorithm was able to provide various kernel arrangements which could potentially explain the observed data as shown in Figure 3.16a and Figure 3.16b.

3.6 Conclusion

In kernel-based regression models, the choice of kernel can greatly impact the predictive performance of the model. As a result, various search methods, such as exhaustive, grid, and randomized search, have been used to find the optimal kernel that explains the given data. Exhaustive search methods tend to offer the best possible combinations, but they are computationally intensive. Grid and randomized search methods can improve efficiency by constraining the search space to a few basis kernel functions, but the time complexity of the search still depends on the nature of the model.

In this chapter, trusting the computational efficiency of the sparse variational Gaussian process, we employed the exhaustive search approach to find a suitable kernel. The algorithm successfully unveiled the hidden patterns within the given data and provided a fitting covariance matrix. However, the approach requires model training and evaluation

for every kernel combination which affects the search efficiency. In addition, the time needed for the ELBO convergence adds an extra time complexity. In spit of that the research also showed the sparse variational Gaussian model outperform LSTM deep learning models in computational efficiency and providing a predictive distribution. However, it suffers from a generalization problem. For data that exhibits multiple periodicity, the approximation favors those features of the data that are bound to happen more frequently. As such, moments which are rare but could happen periodically are ignored and lumped as same with the most frequent moments during forecast. Hence, multiple predictive models with separate data aggregation and analysis could rectify these issues.

For most machine learning algorithm, more training data often entails better approximation. However, for sparse variational Gaussian model, more training data doesn't always mean better approximation. Increasing the training sample while holding the number of inducing variables constant, introduces higher predictive uncertainty. As a result, one could manipulate the sample size, hyperparameter space and kernel setting to find a better coverage probability (PICP) with minimum confidence interval (MPIW). If there is a predictable pattern within the data, including more points on the training phase could be disadvantageous unless more inducing points are used. Since the size of the inducing variables and arbitrary assignment affects accuracy, subjective analysis of the data and determination of optimal inducing location is necessary. So that a compromise can be made on the computational resources one is willing to sacrifice versus the desired performance goal.

To curb the limitation of the SVGP model as a framework for kernel evaluation, we introduced a randomised column sampling technique for a fast predictive model building and proposed a random sparse Gaussian approximation (RSGA) algorithm. We used the variational Gaussian algorithm as a benchmark to test the predictive performance of the suggested algorithm. The experiment has shown that random column sampling offer an alternative method for Gaussian approximation that is scalable to large data. The algorithm follows a stochastic sampling scheme. However, this randomness could be directed and constrained to a subset of the data through frequency sampling. We applied frequency analysis as a mandatory data preprocessing step. This has enabled us to extract points that could potentially summarize the given data. Hence, minimised the space for column sampling and improved the computational efficiency of the underlying model.

The experiments demonstrated that a Gaussian approximation based on random sampling achieved a better predictive performance than the variational counterpart. The RSGA was found to be 10 times faster than the VGA. The experiment revealed inference through variational approach requires a lot of computation. This has affected the computational efficiency of the model. Furthermore, VGA's dependency on the number and location of inducing variable and the type of kernel has affected the predictive accuracy of the model. We have used and enjoyed the benefits of variational inference in other projects. We also know that its predictive accuracy can easily be affected by so many parameters. Although, it provides an ingenious way of dealing with the Bayesian bottleneck, model training requires a lot of tuning and patience to see the desired result. Hence, for a fast and scalable Gaussian inferences, predictive algorithms such as the RSGA can be utilized as an alternative approach for inference and approximation.

In this chapter, we also proposed a stochastic kernel search algorithm that takes into account the limitation of the exhaustive search approach for optimal kernel evaluation. The objective was to utilize the random sparse Gaussian process as the underlying model and provide a framework for efficient estimation of suitable kernels. A combination of sparsity in model building and a stochastic approach for kernel selection has afforded the method a computational advantage over other counterparts. However, it is still stochastic in nature and hence, it requires iterative retraining to find the best combinations.

This approach can be used at the preprocessing stage to determine which combinations of kernels provide better predictive performance and fit the observed values best. On top of that, the algorithm can simplify the application of kernel-based learning to those who find kernel selection vague. To the experienced, it can give a preliminary insight into the structure of the kernels that could potentially fit the data. Regardless of their computational difference, the exhaustive and stochastic kernel search methods can help in retrieving the hidden structures embedded within the given data. Especially to data that exhibit difficult patterns to discern. However, combining kernels in whatever manner increases the complexity of the model and the number of hyperparameters. As such, by minimizing the number of mixtures, in parallel with the desired performance and the available computational resources, the model predictive efficiency can be guaranteed.

Chapter 4

Uncertainty in Deep Learning Models

4.1 Introduction

In chapter 2 and 3, we have reviewed in details the implementation and limitations of gaussian process model. Despite its shortcomings, gaussian model follows a more direct and credible approach to uncertainty quantification in modeling building process. It addresses the epistemic uncertainty as function of the relative distance between the recorded observations within the covariance matrix of the prior distribution. By optimising the number of kernels, their type and hyperparameters, it anticipate and mitigate their impact on predictive accuracy. For the consideration of the inherent and irreducible data variability, the gaussian process assumes an observational noise in the model likelihood function. Yet the exact parametric form of the uncertainty is specific to the problem at hand and varies depending on the posterior predictive distribution. As such, there is some degree subjectivity in the assumed error distribution. On the other hand, Neural networks show a great flexibility in handling big data and making point estimates. However, the lack of uncertainty quantification through a predictive distribution has constrained their application in sensitive areas. Both gaussian and deep learning models have exhibited great performance in providing a flexible function to a given problem. In fact, in previous research it has also been shown that a neural networks with infinite hidden layer will converge to a gaussian prior over functions. Furthermore, for specific priors on network parameters and transfer functions, a fitting covariance functions can be computed analytically, providing a network prediction with infinite hidden layers in a time-complexity of $\mathcal{O}(n^3)$, resembling the gaussian process posterior[106, 107]. Although some equivalence can be established between the two, there is also a great divide based on their parametric nature, the way how they look the observed data, its significance and interpretation, and means of parameter optimisation.

In chapter 2 the gaussian process was defined as a non-parametric regression model. This definition doesn't imply there are no parameters in the gaussian process. However, the non-parametericity signify that the number of parameters required to define a given problem has no relation relation with the amount of data. Hence it doesn't grow or shrink with the size of the data. The parametric profile of a gaussian process is fixed regardless of the size of the data. Consequently, it is fully specified by a fixed set of parameters. More precisely, the mean, covariance function and the hyper-parameters of the kernel function. On the other hand, deep learning models are parametric. Their flexibility to generate complex patterns commensurate with the given problem emanates from their tendency to expand their parameter space and accumulate as much parameter as required. This often times leads to overfitting, poor predictive performance and over-

parameterised model. The difference in the parameterisation of the models has affected how the models see the observed data. For a gaussian process, in addition to optimising the hyper-parameters, the observed data determines the future outcomes of the model. This is evident in chapter 2, parameter learning in equation (2.41) and posterior mean in equation (2.40). Hence, the observed data impact the present form and future of behaviour of the model. Consequently, the observed data needs to be kept and monitored for future reference thereby causing a huge storage constraints. In case of deep learning and other parametric models as well, the reach of the observed data extends up to modifying the parameters of the model. Hence, it doesn't directly affect the outcome of the model during prediction. Consequently, it can be discarded after model training or parameter optimisation. In spite of their differences, as predictive model, the quantification of uncertainties involved during prediction is crucial and the objective assessment of these uncertainties improves the validity of the model. Researches in the areas of probabilistic deep learning has allowed the predominately mean centered network estimation to have the ability to anticipate variability and provide a predictive distribution.

This chapter proposes a new approach for a predictive distribution in deep learning models. It will focus on ensuring the quantification of both sources of uncertainties in a way that provides a minimum confidence interval with maximum coverage for prediction points. We will demonstrate mathematically, the upper lower bound assessment method encompassing the quality metrics, prediction interval coverage probability (PICP) and mean prediction interval width (MPIW) in its bound estimation. Then, we developed a customized loss function with adaptive hyperparameter that balances the needed coverage probability in relation to the prediction interval. Finally, we evaluated the performance of our approach on a UCI regression data using the recent Quality Driven (QD) bound estimation method as a benchmark.

4.2 Uncertainty Estimation in Deep learning

Uncertainty quantification is a hallmark of a predictive model. Even though there are areas that don't require it (i.e joint torque estimation for a robotic system [108]), the majority relay on uncertainty estimation for decision making [91]. For example, the range of uncertainties is used for cost profit optimization in energy and stock market regression models. In the case of a classification oriented tasks such as AI assisted clinical decision making [109], post office and nuclear power plant [110], a probabilistic model output with high uncertainty could invite a second opinion from experts for better interpretation and improved judgment. Therefore, the inclusion of uncertainty during prediction enhances the acceptability and validity of the model [111].

Deep neural nets have demonstrated impressive predictive agility. They have been used for regression and classification based predictions in a variety of areas [76]. Medical [109], weather [111], transportation [112], energy [113], stock [114] and robotics [108], are some of the specialization where deep neural net is gaining momentum. However, the lack of uncertainty estimation has restricted their application in sensitive areas [109, 115]. This can be attributed to the absence of probabilistic modeling which has limited their outputs to point estimations in case of regression and labels without confidence in case of classification. A variety of approach have been proposed to establish a probabilistic output distribution for neural nets. Deep ensembles, distribution based [115, 116] and distribution free interval estimation [117, 118], Bayesian inference [119], probabilistic irregularity through MC dropouts [110] are few notable mentions.

For machine learning uncertainty quantification, understanding the source of the uncertainty is important. The uncertainties can emanate inherently from the model or could

be external to it and part of the input observation. Predictive models optimize their parameters based on the quality and quantity of data in which they are permitted to train. Their generalization about the problem at hand is directly driven from their experience. As such, epistemic uncertainty arises when either the quantity or the quality of the data is low enough to affect its inference ability. As a result, they are also called knowledge based uncertainties. They can arise due to the train-test data split irregularity that skew or bias the training data on which the machine is being trained. This creates a difference between the training and validation data forcing the model to be trained on a wrong and non-representative data which inadvertently affects the inferential ability of the underlying model. Model misconfiguration, inadequate size and quality of the data are also some other factors that constrain the knowledge building process and contribute to presence of epistemic uncertainty [120, 76]. Meaning that by providing enough data with a higher representative information, avoiding sampling bias during training point selection and improving the quality of the data, epistemic uncertainties can be mitigated and reduced.

On the other hand, uncertainties due to the randomness or variability of the input data are called aleatoric uncertainties. All data that is ever recorded contains a bit of randomness or noise. Unlike the epistemic uncertainty, improving the quantity or quality of data doesn't help in mitigating aleatoric uncertainties. If every bit of data that is collected comes with a noise, collecting some more data won't help in minimizing the effect of these uncertainties. It is one aspect of the data that either the human observer or the machine has no control over. As such, it is not the property of the model, rather the intrinsic property of the data. Consequently, it is also called the data variance. Owing to its stochastic nature and dependence on the observations rather than the model, anticipating and rectifying aleatoric uncertainty is challenging. No amount of new information or model configuration will help in reducing aleatoric uncertainties. They are the irreducible noises embedded within the input data. They are mitigated through a probabilistic distribution by assuming a conditional probability of the output on the input along with all the parameters affecting it [120, 76].

Probabilistic models like the gaussian process account for data uncertainty by assuming the observed values as samples drawn randomly from a stochastic process defined by $y(x) \sim \mathcal{N}(\mu_x, \Sigma_{xx} + \sigma^2 I_n)$, where σ^2 represent the anticipated data variability. In image processing, data augmentation and transformation techniques (i.e rotation, translation, flipping, cropping, etc) are well know techniques that have been used widely to minimize the effect these uncertainties which leads to a robust predictive performance during model development phase. Especially, in the presence of less data, data augmentation on the training data could help in improving the generalization capability of the model to instance which the model has never seen or trained before. During validation the same approach can also be used to populate the validation data with auxiliary test point so that the model can be evaluated more robustly.

A number of techniques have been devised to account for the two uncertainties in neural net prediction. Some, like the Bayesian neural net (BNN) follow a principled mathematical formulation to estimate the epistemic uncertainty. However, their implementation imposes a huge computational requirement [109, 115]. A distribution free interval estimation algorithm such as the Quality Driven [118], offers a higher computational efficiency and a simplistic modeling that incorporate the quality metrics into the interval estimation. It accounts epistemic uncertainty through an ensembles of neural net models. However, it fails to provide a minimized prediction interval (PI) for high frequency data. On the other hand, the distribution based interval estimation methods, assumes a conditional probability distribution on the input data. This approach creates a mathematical convenience to address the aleatoric and epistemic uncertainties. However, it fails to incorporate the quality metrics (i.e PICP and MPIW) into consideration. Hence,

for a successful bound estimation through a distribution based approach, the selection of the error distribution should complement the metrics that is used to evaluate the performance of the predictive model. In [115, 116], a gaussian error distribution is assumed on the input data, and now if MPIW is the intended metrics that is employed to asses model performance, a gaussian error distribution, although acceptable, is not the right distribution. We believe a logistically approximated gaussian distribution will simplify the assimilation of the quality metric MPIW in PI optimization in a meaningful manner as compared to other distributions.

In neural network, more heuristic approach to predictive distribution is through point estimate aggregation. It should be noted that in chapter 3, we followed a similar method and trained 21 LSTM neural networks in generating the electricity demand for two weeks ahead predictive distribution. The major challenge in training multiple models to acquire and anticipate data variability through a distribution is that, the time required for model training scales linearly with the number of models considered. On top of that, deep learning models generalization ability puts the validity of this approach into question. For example, even with different network setting and parameter initialization, the network parameters are bound to converge and provide a similar predictions. As a result, the approach doesn't provide a guarantee in the fidelity of the estimated coverage probability and confidence interval. An intuitive yet straightforward approach to rectify the network predictive convergence is suggested through the introduction of a probabilistic dropout layer [76, 110, 121]. This method employs dropping a percentage of the networks node stochastically in manner similar to Monte Carlo (MC) sampling during training. It assigns a Bernoulli random variable z_{ij} with a probability of success p_i to every input point j in every network layer i. The unit j in layer $i-1$ will be dropped as an input to layer i if its corresponding random variable $z_{ij} = 0$. As such, not all aspect of the data is visible to all nodes of the network. This artificial mask creates a self induced randomness in model's prediction. Thereby providing a means for uncertainty quantification and at the same time penalizing over-fitting [76, 110].

A more valid and principled approaches to uncertainty quantification has been forwarded through the Bayesian framework. The Bayesian neural net (BNN), assumes a prior $p(w|\theta)$ on the networks parameters. Then, given the observations ($\mathcal{D}$), a Bayesian inference is made to compute the posterior probability of the weights ($p(w|\theta, \mathcal{D})$) which will later be used to find the predictive distribution of the network output [76, 109, 115]. Unfortunately, this results in an intractable posterior computation. To address it, an approximate posterior evaluation is suggested using sampling (i.e MCMC). Bayesian neural net took the predominately point estimation to a distribution, despite the intractability of the posterior evaluation and the slow MCMC algorithm which resulted in a higher computational cost [76, 119, 115]. In an effort to rectify this, an approximate posterior evaluation is suggested through a variational distributions and MC dropout [110, 122]. The variational inference takes the Bayesian posterior as an optimization problem and try to approximate it through a variational distribution that is more manageable to numerical methods [122]. Whereas, MC dropout creates a probabilistic irregularity on the number of network weights considered for output evaluation. This approach has been mathematically proven to provide an approximate posterior in deep ensembles [110]. Mean variance estimation (MVC) for deep ensembles [123] offer yet another approach to a predictive distribution. This distribution based estimation method apply the likelihood criterion as an objective function and utilize a neural network with two outputs to predicts the mean and variance of the target distribution. For uncertainty estimation, it apply an ensemble learning with randomization based approach where each member of the ensembles is trained on a random parameter initialization and data shuffling to provide an aggregated predictive distribution. As such, it captures the aleatoric uncertainty through a gaussian

target-error distribution and minimize the epistemic using the aggregated ensembles [123].

The error based uncertainties quantification techniques mentioned above mostly gravitate toward maximizing the coverage probability of points closer to the mean trajectory. Even then there is no guarantee to the optimality of the prediction interval (PI) or the proportion of points enclosed within the margin of doubt. This has been one of the challenging problem in uncertainty quantification both in parameter and non-parametric models. The lower upper bound estimation (LUBE) [20] and Quality driven PI [118] estimation methods, perhaps can be considered as a breakaway from the traditional error-based interval estimation methods. These methods focus on maximizing the probability of the prediction interval in containing the epistemic and aleatoric variability of the individual data point while at the same time minimizing the constructed interval width. To that end, they followed a distribution-free estimation method with a new criterion to asses the quality of the prediction interval. They argued that the interval width and its coverage probability should be the proper criterion by which the quality of prediction interval should be measured. As such, the mean prediction interval width (MPIW) and prediction interval coverage probability (PICP) are employed to objectively quantify the optimality of the constructed PI. LUBE develops a combinational width-based criterion that provides a narrow PI while maintaining optimal coverage probability commensurate with the performance metrics. However, the LUBE loss function doesn't integrate well with gradient based optimization techniques. The Quality Driven distribution-free algorithm (QD) builds upon LUBE and address its limitation by proposing an alternative cost function that is compatible with gradient based optimizers. It provides an elegant approach to PICP and MPIW modeling for PI minimization. The algorithm is simple and yet with faster convergence time and accurate prediction. However, QD and LUBE fail to provide a minimized PI for data with high frequency and complex pattern. For such data, they fulfill the needed coverage probability at the expense of a higher PI width.

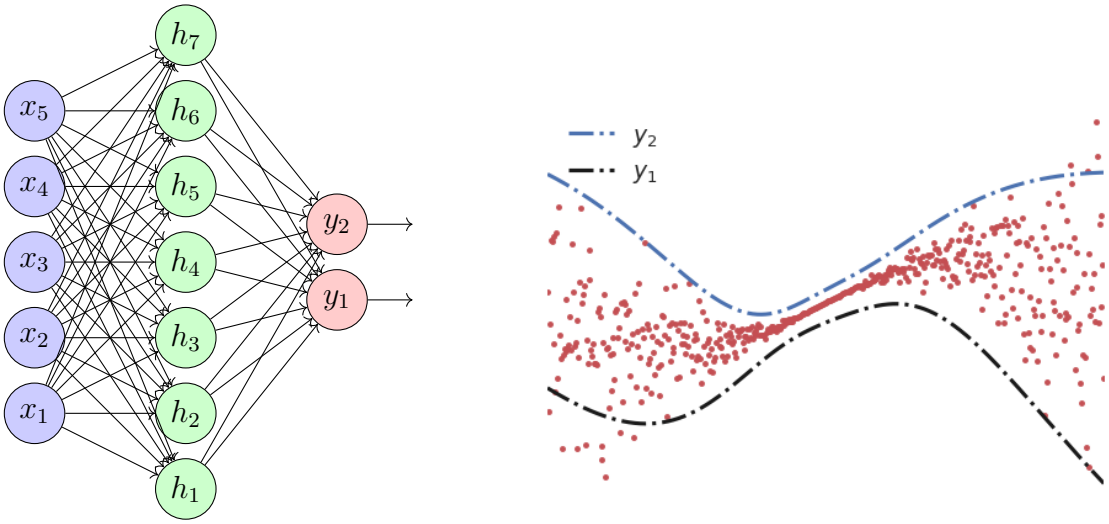
The main argument in the distribution-free estimation methods such as LUBE [20] is that, the quality of the PI in distribution-based estimation methods is questionable for the very reason they do not consider the quality metrics (i.e PICP and MPIW) in their derivation. In this Chapter, we will provide a mathematical proof that a distribution-based estimation methods could incorporate the quality metrics towards an algorithm that provides a minimized PI with high coverage probability for low and high frequency data. Our objective is to propose a distribution based lower upper bound estimation technique. The proposed cost function can be minimized by any of the available gradient based optimization algorithm. We will test the algorithm on a synthetic and real data. As a benchmark, the algorithm will be compared with the latest QD algorithm.

4.3 Lower Upper Bound Estimation

Like the MVC model, the lower upper bound estimation model follow the same network architecture shown in Figure 4.1, with two output nodes y_1 and y_2 at the output layer. The only exception being, in the MVC, the outputs y_1 and y_2 represent the estimated mean and variance of the target distribution. In case of LUBE, they represent the estimated lower and upper bounds as shown in Figure 4.1-(b) respectively. Assuming we are given a data such as shown in Figure 4.1, the central theme in LUBE is to maximize the probability of finding individual data points y_i between y_1 & y_2 and minimize the average distance between the two

$$\arg\min \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_{2,i} - y_{1,i}) \right\} \tag{4.1}$$

Furthermore, assuming a gaussian error distribution and an independent data distri-



(a) Network graph (b) Lower upper bound estimation

Figure 4.1: General LUBE network graph with two outputs and sample bound estimation

bution, the data points can be modeled with random variable $y(x) = f(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ or equivalents as $y(x) \sim \mathcal{N}(\mu_x, \sigma^2 I_n)$ where σ^2 represent the anticipated data variability. Now, for any data point y, the probability density is given by

$$p(y) = \frac{1}{\sqrt{2\pi}\sigma} exp\Big(- \frac{(y - \mu_x)^2}{2\sigma^2}\Big) \tag{4.2}$$

And the probability it lies within the bound y_1 and y_2 is given by

$$\begin{aligned} & p(y_1 < y < y_2) = \Phi\Big(\frac{y_2 - \mu_x}{\sigma}\Big) - \Phi\Big(\frac{y_1 - \mu_x}{\sigma}\Big) \\ & \text{where} \\ & \Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} exp\Big\{ - \frac{1}{2}u^2\Big\} du \end{aligned} \tag{4.3}$$

In equation (4.2) & (4.3), the relative distance between the data point and the mean of the distribution is given as a squared difference. In addition to the inconvenience of evaluating $\Phi(y)$, this representation will minimize the squared deviation, not the average mean deviation given in equation (4.1). In a distribution-based interval estimation methods such as in [115, 116, 123], considering a gaussian error distribution the assumption, contrary to the actual PI, it mostly minimizes the squared distance between the prediction bounds and the mean of the distribution. Consequently, we considered a logistically approximated gaussian data distribution. This approximation creates a mathematical convenience to assimilate the quality metrics directly into the modeling and optimization, along with the familiarity of gaussian-like error distribution for aleatoric uncertainty quantification.

A gaussian distribution with $\mathcal{N}(\mu, \sigma^2)$ can be approximated by a logistic distribution with $\mathcal{L}(\mu, \frac{\pi\sigma}{3})$ as shown in Figure 4.2. As such, for the prediction interval (MPIW) minimization, we considered a general logistic data distribution $\mathcal{L}(\mu, s)$ with probability density function (PDF) for any random variable Y given by

$$p(\mathrm{Y} = y) = \frac{e^{-(y-\mu)/s}}{s(1 + e^{-(y-\mu)/s})^2} \tag{4.4}$$

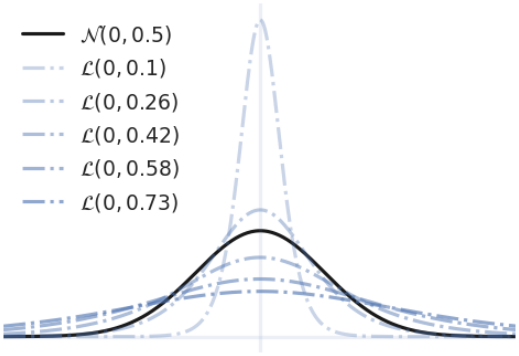


Figure 4.2: Logistic-gaussian approximation

and a cumulative distribution function (CDF)

$$p(\mathrm{Y} \leq y) = \frac{1}{1 + e^{-(y-\mu)/s}} \tag{4.5}$$

Assuming a network prediction $\{y_1, y_2\} \epsilon Q$, where Q represent the space of all possible bound functions, and $y_1 \leq y \leq y_2$, the equation for PI optimization can be derived from the CDF in equation 4.5. Unfortunately, interval minimization using CDF is not computationally stable. The reason being the success of PI minimization through CDF is predicated upon the assumption that the network always forecast $y_1 \leq y \leq y_2$, an assertion which might not be true all the time. A more computationally stable interval minimization (i.e MPIW) can be derived using the PDF given in equation 4.4. For the coverage probability maximization, we followed a simple membership assignment for the absence and/or presence of the observation within the prediction bound. That means, the PI optimization problem can be broken down and discussed separately as the minimization of MPIW and maximization of PICP. Later on, a custom loss function will be derived by combining the interval minimization and coverage probability maximization.

4.3.1 Interval width minimization

By definition, the width of the prediction interval is quantified through the mean prediction interval width (MPIW)[20] which is given by

$$\mathrm{MPIW} = \frac{1}{n}\sum_{i=1}^{n}(y_{2,i} - y_{1,i}) \tag{4.6}$$

where n is number of observations and $y_{1,i}$, $y_{2,i}$ are the lower and upper bounds of the predicted PI for the i^{th} sample respectively. Now, for a time series data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, the prediction bounds are assumed to be samples drawn from a logistically distributed random variable Y

$$Y = f(x_i) + \epsilon, \qquad \epsilon \sim \mathcal{L}(0, s = 1) \tag{4.7}$$

$$Y \sim \mathcal{L}(f(x_i), s = 1) \tag{4.8}$$

where $f(x_i) = y_i$ and s are the mean and scale of the distribution.

For a logistically distributed random variable $Y \sim \mathcal{L}(y_i, s = 1)$, and $\{y_1, y_2\} \epsilon \mathrm{Y}$, we compute the PDF of the lower y_1 and upper bound y_2 is given as

$$p_{2,i}(Y_i = y_{2,i}) = \frac{e^{-(y_{2,i}-y_i)/s}}{s(1 + e^{-(y_{2,i}-y_i)/s})^2} \tag{4.9}$$

$$p_{1,i}(Y_i = y_{1,i}) = \frac{e^{-(y_{1,i}-y_i)/s}}{s(1 + e^{-(y_{1,i}-y_i)/s})^2} \tag{4.10}$$

where $Y_i \epsilon Y$. Applying a log transformation on the LHS & RHS of equation (4.9) - (4.10) and simplifying it

$$\begin{aligned} y_{2,i} &= y_i - s\ln s - s\ln p_{2,i} - 2s\ln\left(1 + e^{-(y_{2,i}-y_i)/s}\right) \\ y_{1,i} &= y_i - s\ln s - s\ln p_{1,i} - 2s\ln\left(1 + e^{-(y_{1,i}-y_i)/s}\right) \\ y_{2,i} - y_{1,i} &= s\ln\left(\frac{p_{1,i}}{p_{2,i}}\right) + 2s\ln\left(\frac{1 + e^{-(y_{1,i}-y_i)/s}}{1 + e^{-(y_{2,i}-y_i)/s}}\right) \end{aligned} \tag{4.11}$$

Equation (4.11) provides the interval width for a single data point. Applying equation (4.11) for all points in our data set and computing MPIW using equation (4.6)

$$\begin{aligned}
\text{MPIW} &= \frac{1}{n}\sum_{i=1}^{n}(y_{2,i} - y_{1,i}) \\
\text{MPIW} &= \frac{1}{n}\sum_{i=1}^{n}\left\{ s\ln\left(\frac{p_{1,i}}{p_{2,i}}\right) + 2s\ln\left(\frac{1+e^{-(y_{1,i}-y_i)/s}}{1+e^{-(y_{2,i}-y_i)/s}}\right)\right\} \\
&= \frac{s}{n}\sum_{i=1}^{n}\left\{ \ln\left(\frac{p_{1,i}}{p_{2,i}}\right) + 2\ln\left(\frac{1+e^{-(y_{1,i}-y_i)/s}}{1+e^{-(y_{2,i}-y_i)/s}}\right)\right\}
\end{aligned} \tag{4.12}$$

We can minimize equation (4.12) by maximizing equation (4.9) and (4.10). As we maximize the individual PDFs, the predicted bounds will approach to the data mean trajectory which will drive the second term in RHS of equation (4.12) to zero. Hence, PI optimisation can be carried out by maximizing the two PDF's. Consequently, the prediction interval minimization can formally be defined as

$$\begin{aligned}
\arg\min\{\text{MPIW}\} &= \arg\min\left\{\frac{s}{n}\sum_{i=1}^{n}\left\{ \ln\left(\frac{p_{1,i}}{p_{2,i}}\right) + 2\ln\left(\frac{1+e^{-(y_{1,i}-y_i)/s}}{1+e^{-(y_{2,i}-y_i)/s}}\right)\right\}\right\} \\
&= -\left\{\arg\max\left\{\frac{s}{n}\sum_{i=1}^{n}\left\{ \ln\left(\frac{p_{1,i}}{p_{2,i}}\right) + 2\ln\left(\frac{1+e^{-(y_{1,i}-y_i)/s}}{1+e^{-(y_{2,i}-y_i)/s}}\right)\right\}\right\}\right\} \\
&\approx -\left\{\arg\max\left\{\frac{s}{n}\sum_{i=1}^{n}\left\{p_{1,i}+p_{2,i}\right\}\right\}\right\}
\end{aligned} \tag{4.13}$$

The last assertion in equation (4.13) is the consequence of the logarithm function being a strictly increasing function. However, for numerical stability or to avoid adding and dividing small numbers, the logarithmic version in equation (4.13) can also be used.

4.3.2 PICP maximization

The quality metric PICP quantifies the percentage of points falling within the PI. To that end, we attached a binary variable $\phi_i \in \Phi_i$ that takes the values of 1 and 0 with probabilities $(1-\alpha)$ where $\alpha = 0.05$ to each data point, signifying the absence and/or presence of the point within the interval. Each observations of the random variable Φ_i are defined as

$$\phi_i = \begin{bmatrix} 1, & y_{1,i} \le y_i \le \hat{y}_{2,i} \\ 0, & otherwise \end{bmatrix} \tag{4.14}$$

For a single observation, the random variable Φ_i follows a Bernoulli distribution given by

$$p(\Phi_i = \phi_i) = (1-\alpha)^{\phi_i} * \alpha^{(1-\phi_i)} \tag{4.15}$$

Assuming c is the total number of points covered under the PI, and given n observations that are iid and having the same probability of success $(1-\alpha)$, the random variable Φ_i follows a Binomial distribution given by

$$p(\Phi_i = c) = \binom{n}{c}(1-\alpha)^{\phi_i} * \alpha^{(n-\phi_i)} \tag{4.16}$$

PICP modeling and maximization through Binomial distribution is elegantly demonstrated in [118]. In this chapter, however, we follow a rudimentary approach in evaluating

and optimizing PICP. By definition [20, 118],PICP measures the percentage of points convered under the PI prescribed by a confidence limit ((1 − α). From equation (4.14), the total number of points covered within the PI can be estimated as

$$c = \sum_{i=1}^{n} \phi_i \tag{4.17}$$

Consequently, the coverage probability can be computed as

$$\text{PICP} = \frac{c}{n} \tag{4.18}$$

For most application, a coverage probability PICP $\geq (1-\alpha) * 100\%$ is required. Hence, the PICP maximization can formally be defined through a constraint

$$\arg\max \Big\{ \text{PICP} - (1-\alpha) \geq 0 \Big\} \tag{4.19}$$

Hence, a maximized PICP can be found by minimizing equation (4.19)

$$\arg\min \Big((1-\alpha) - \text{PICP} \Big) \tag{4.20}$$

Combining the MPIW and PICP optimization problems stated in equation (4.13) and (4.20), a general objective function $\mathcal{L}$ can be defined as

$$\begin{aligned} \mathcal{L} &= \arg\min \Big\{ \text{MPIW} \Big\} + \arg\max \Big\{ \text{PICP} \Big\} \\ &\quad - \Big\{ \frac{s}{n} \sum_{i=1}^{n} \Big\{ p_{1,i} + p_{2,i} \Big\} \Big\} + \Big\{ (1-\alpha) - \text{PICP} \Big\} \\ \mathcal{L} &= -\frac{\lambda_1}{n} \sum_{i=1}^{n} \Big(p_{1,i} + p_{2,i} \Big) + \lambda_2 \Big((1-\alpha) - \text{PICP} \Big) \end{aligned} \tag{4.21}$$

Where the parameters λ_1 and λ_2 are design parameters which will generalize the loss function depending on the data distribution considered and the order of priority given to the quality metrics. For instance, λ_1 can be made an adaptive parameter based on the the values of PICP to set the variance of distribution so that the network will be forced to sample from a specific distribution. On the other hand, the parameter λ_2 can be used to establish a preferential priority on which quality metric (i.e prediction width or coverage probability) the network should focus on optimizing. Especially, for data with asymmetric distribution, tuning the values of λ_2 improves the accuracy of prediction. For gaussian-like data distribution the parameter λ_2 can be ignored or assigned a value $\lambda_2 = 1$.

One of the challenges in distribution based bound estimation models is finding a suitable value for the distribution variance. The width of the interval and coverage probability have an inverse relationship during the optimization. Coverage probability (i.e PICP) should be maximized while minimizing the coverage width (i.e MPIW). This inverse relationship can be modeled by a decaying exponential function. As such, the parameter λ_1 can be turned into an adaptive parameter that regulate the variance of distribution based on the current values of the PICP

$$\lambda_1 = e^{-\beta * \text{PICP}} \tag{4.22}$$

where $\beta > 0$ is a design parameter for the exponential decay rate. Such assignment will force the network to sample candidate bound values from a distribution with a variable variance depending on the value of the PICP.

In equation (4.14) and (4.17), the existence of a data point within the PI and the total number of data point is evaluated respectively. However, a step function membership assignment in equation (4.14) for PICP evaluation creates computational inconvenience when minimizing the loss. As such, an alternative differentiable approximation suggested in [118, 124] is adopted. Hence, equation (4.14) and (4.17) are modified accordingly

$$\Big[k_{up}\Big] = \begin{bmatrix} \frac{1}{1+e^{-\gamma(y_{2,i}-y_i)}}, & y_i \leq y_{2,i} \quad i = 1, \cdot\cdot, n \\ 0, & otherwise \end{bmatrix} \tag{4.23}$$

$$\Big[k_{lo}\Big] = \begin{bmatrix} \frac{1}{1+e^{-\gamma(y_i-y_{1,i})}}, & y_{1,i} \leq y_i \quad i = 1, \cdot\cdot, n \\ 0, & otherwise \end{bmatrix} \tag{4.24}$$

Then the total number of data points inside the PI can be given as

$$c = k_{up} * k_{lo} \tag{4.25}$$

4.3.3 Custom loss algorithm

The summary of the proposed distribution based interval optimisation is demonstrated in algorithm 4. The network will be trained with the objective function defined in equation (4.21). To that end, a custom loss function is written as shown in algorithm 4. Given the data and user-defined parameters, the algorithm will adaptively set the parameter λ_1 based on the current coverage probability. In the algorithm, the gain parameter λ_2 can be set to 1 for a normal-like distributed data. However, a value different $\lambda_2 \neq 1$ will improve the PI optimization for data with asymmetrical distribution. Computing the PDF's can introduce smaller number that might lead to overflow and numerical instability, if that happens the log-PDF's can be applied.

Algorithm 4: Distribution based LUBE loss

Input: β, γ, λ_2, true value (y_t), upper bound (y_2), lower bound (y_1)
if $y_2 \geq y_t$ **then**
  $k_{up} = sigmoid\Big(\gamma(y_2 - y_t)\Big)$
end
if $y_1 \leq y_t$ **then**
  $k_{lo} = sigmoid\Big(\gamma(y_t - y_1)\Big)$
end
$\text{PICP} = \frac{1}{n} * (k_{up} * k_{lo})$
$\lambda_1 = exp\Big\{ -\beta * \text{PICP} \Big\}$
$p_2 = p(y_2 = y_t)$
$p_1 = p(y_1 = y_t)$
$MPIW = \frac{1}{n}(p_2 + p_1)$
$loss = -\lambda_1(\text{MPIW}) + \lambda_2((1-\alpha) - \text{PICP})$
return $loss$

4.4 Uncertainty Quantification

The aleatoric uncertainty has already been defined as the irreducible data noise variance. Regardless, it is a variance that should be counted when considering the total uncertainty.

Together with the model epistemic uncertainty, the total system variance can be defined as

$$\sigma^2 = \sigma_a^2 + \sigma_e^2 \tag{4.26}$$

where σ_a and σ_e are the aleatoric and epistemic uncertainty respectively. The epistemic uncertainty is related to the networks parameter uncertainty. As such, the aggregated ensemble models technique suggested in [118, 123] is applied to mitigate the epistemic uncertainty. That means, a number of models will be trained on random parameter initialization and their bound estimation will be aggregated to provide the network final interval estimates as a distribution. Assuming a uniform mixture weight for each model in ensemble and given the upper (y_2) and lower (y_1) bound estimates of the z models in ensemble, the aggregated distribution is given by

$$y_2 \sim \mathcal{N}\Big(\mu_2 = \frac{1}{z}\sum_{i=1}^{z} y_{2,i},\ \sigma_2^2 = \frac{1}{z-1}\sum_{i=1}^{z}(y_{2,i} - \mu_2)^2\Big) \tag{4.27}$$

$$y_1 \sim \mathcal{N}\Big(\mu_1 = \frac{1}{z}\sum_{i=1}^{z} y_{1,i},\ \sigma_1^2 = \frac{1}{z-1}\sum_{i=1}^{z}(y_{1,i} - \mu_1)^2\Big) \tag{4.28}$$

where $y_{2,i}$ and $y_{1,i}$ describe the upper and lower bound estimates of model i. Now, following equation (4.27) and (4.28) and assuming a 95% coverage probability, the final upper and lower bound network predictions for ensembled models ca be given as

$$\begin{aligned} y_2 &= \mu_2 + 1.96 * \sigma_2 \\ y_1 &= \mu_1 - 1.96 * \sigma_1 \end{aligned} \tag{4.29}$$

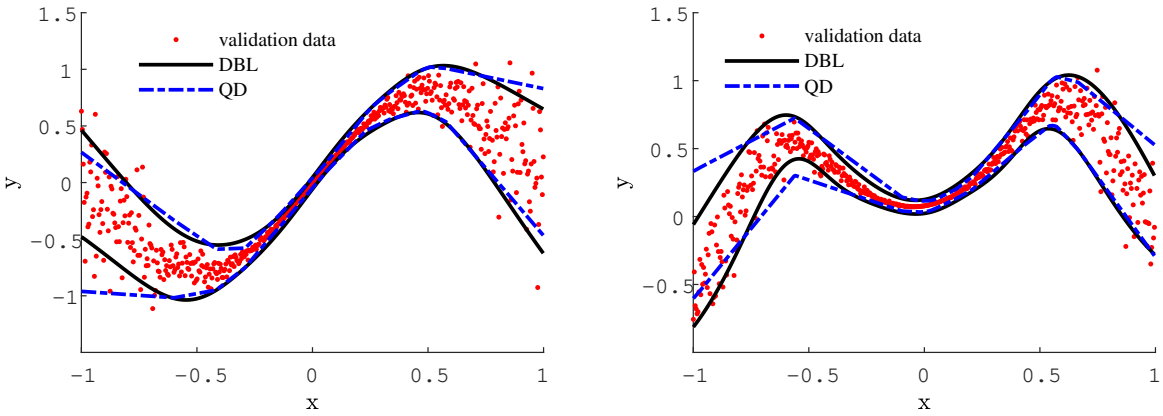
where y_2 and y_1 are the forecasted upper and lower bounds of the network.

4.5 Distribution based lower upper bound simulation

The proposed distribution-based lower upper bound estimation (DBLUBE) algorithm was tested on a synthetic and real data. To account for the trend, seasonality and cyclic patterns that is exhibited in a time series data, the synthetic data was formed as a combination of polynomial and trigonometric parametric equation. The data variability is simulated using a gaussian and exponential distribution.

$$\begin{aligned} y_1 &= 2sin(2.5\pi x^2) + x + x^2 + \epsilon \\ with\ \epsilon &\sim \mathcal{N}(0, x^2) \\ and\ \epsilon &\sim Exp\Big(\frac{1}{x^2}\Big) \end{aligned} \tag{4.30}$$

The predictive performance of DBLUBE is evaluated using as a benchmark the recent quality driven (QD) interval estimation algorithm suggested in [118] for deep neural networks. The difference between the two algorithms is that DBLUBE takes a prior assumption on the distribution of the given data. However, their tendency to provide prediction based on the same quality metrics and in terms of PI bounds makes them similar. A number of models were trained using the two algorithms under the same network architecture and on a 500 randomly sampled data points generated according to $y_2 = 2sin(\pi x) + \epsilon$ and $y_3 = 2sin(1.5\pi x^2) + x + x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, x^2)$ and $x \sim \mathcal{U}[-1, 1]$. The experiment showed that for a low frequency data, both algorithms score a similar performance index in terms of the required PICP quality metrics as shown in Figure 4.3-(a). However, for high frequency data, without adding extra hidden nodes, the QD's ability to provide a minimized MPIW diminishes. It provides a maximized coverage at the expense of a wide



(a) Model prediction with $\lambda = 0.001$ and $\beta = 1$ (b) Model prediction with $\lambda = 0.01$ and $\beta = 1$

Figure 4.3: Single model PI prediction

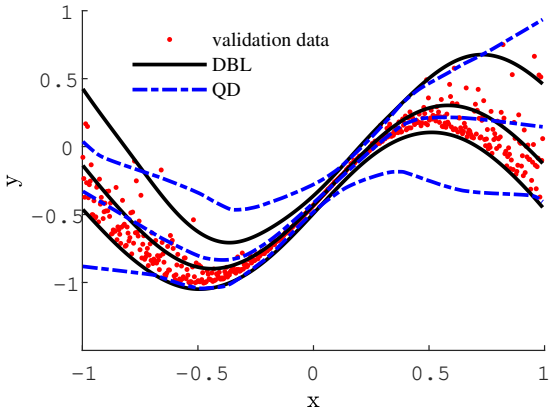
prediction interval as shown in Figure 4.4-(c-d). The simulations shows that DBLUBE has a faster convergence rate and faster computation than QD, both for low and higher frequency data. The optimality of QD's PI is highly dependent on the value of the gain parameter λ. Hence, it requires continuous tuning as the data changes. On the other hand, DBLUBE also needs a good β parameter setting for a faster convergence and optimal quality score. However, the fact that it is distribution based makes its PI prediction less sensitive to changes in data pattern and parameter settings. In Figure 4.3-(a-b), the QD algorithm requires re-tuning its λ parameter as the data frequency changes for minimized PI and optimal coverage. In contrary, DBLUBE achieves an optimal PI and high coverage score for a constant β value and without the need for a re-tuning its parameter.

Table 4.1: QD algorithm performance metrics on real data

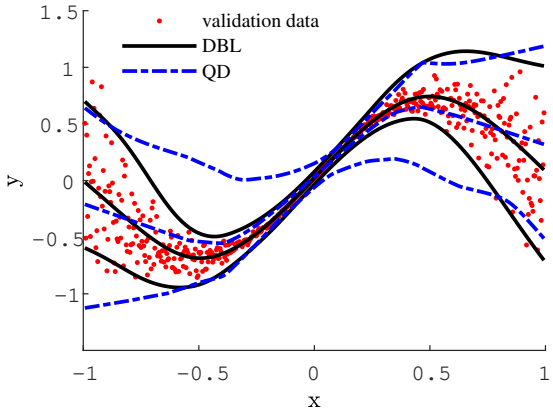
DATA	PICP	MPIW	R^2	RMSE	CVRMSE
Air Quality Benzen Cons.	**0.96±0.02**	**0.65±0.10**	−0.31	0.375	0.622
Airfoil Self-Noise Presu.	0.99±0.00	1.31±0.03	0.504	0.259	1.777
Boston Housing	0.94±0.00	1.10±0.02	0.639	0.239	0.971
Wave Energy Conversion	**0.98±0.01**	**0.50±0.05**	0.920	0.081	0.676
Steel factory E.Consum.	**0.97±0.01**	**2.20±0.21**	0.750	0.213	0.314
Concrete Comp-Strength	0.95±0.00	1.06±0.06	0.659	0.238	1.522
Parkinsons Monitoring	0.96±0.00	2.21±0.15	0.057	0.426	4.942
Forest Fire Area Estim.	**0.98±0.01**	**1.94±0.22**	$***$	0.271	0.278

Note: The actual Data has been normalized in the range (-1,1). As such, the metrics (i.e MPIW, RMSE and CVRMSE) are computed based on this range.

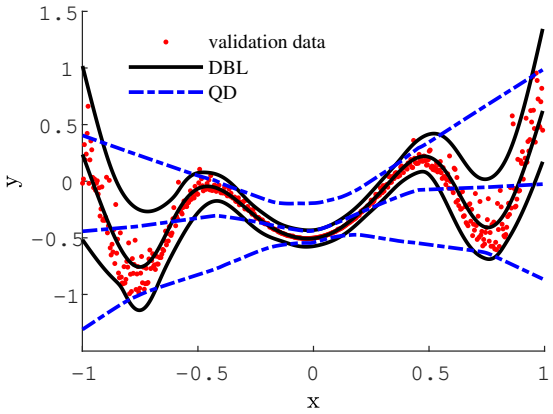
The final PI predictive distribution that takes into account the total model uncertainty was carried out quantitatively by aggregating the predicted PI bounds of multiple neural models according to equation (4.27) and (4.28). Fifteen ensemble models were trained on a randomly generated data (x, y_1) and (x, y_2) using QD and DBLUBE algorithm as a loss function and on a parameter range $\lambda \in [0.01, 0.15]$ and $\beta \in [0, 1.5]$ respectively. The final prediction bounds is evaluated according to equation (4.29). The experiments showed that for a specifically tuned value of λ, QD handles exponential noise better than DBLUBE. However, when varying the λ parameter for the ensemble learning, the final aggregated PI distribution results in a broader MPIW as shown in Figure 4.4-(a-b). On the other hand, the aggregate distribution in equation (4.29) gives DBLUBE enough flexibility and variance to handle exponential noises. Moreover, for data with normal distribution, the



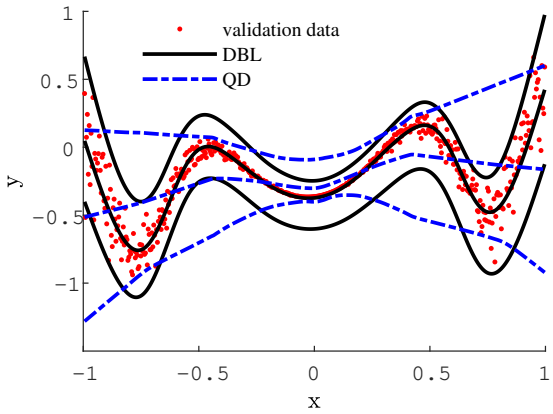
(a) Predictive distribution with $\epsilon \sim Exp(\frac{1}{x^2})$



(b) Predictive distribution with $\epsilon \sim \mathcal{N}(0, x^2)$



(c) Predictive distribution with $\epsilon \sim Exp(\frac{1}{x^2})$



(d) predictive distribution with $\epsilon \sim \mathcal{N}(0, x^2)$

Figure 4.4: Ensemble PI predictive distribution

Table 4.2: DBLUBE performance metrics on real data

DATA	PICP	MPIW	R^2	RMSE	CVRMSE
Air Quality Benzen Cons.	0.99±0.00	0.75±0.15	0.918	0.094	0.156
Airfoil Self-Noise Presu.	**0.98±0.03**	**1.12±0.06**	0.689	0.205	1.406
Boston Housing	**0.96±0.01**	**0.94±0.03**	0.769	0.191	0.776
Wave Energy Conversion	**0.99±0.01**	**0.52±0.05**	0.964	0.053	0.451
Steel factory E.Consum.	0.91±0.00	1.76±0.21	0.762	0.208	0.307
Concrete Comp-Strength	**0.96±0.00**	**1.02±0.04**	0.751	0.200	1.300
Parkinsons Monitoring	**0.95±0.01**	**1.80±0.11**	0.162	0.402	4.661
Forest Fire Area Estim.	0.94±0.01	1.71±0.29	$****$	0.279	0.286

Note: The actual Data has been normalized in the range (-1,1). As such, the metrics (i.e MPIW, RMSE and CVRMSE) are computed based on this range.

means of the DBLUBE's lower and upper distributions provides the needed minimized MPIW while meeting the required coverage probability, without utilizing the variance of the distribution for a 95% uncertainty coverage.

Later on, the performance of the two algorithms were evaluated on real data from UCI repository. Most of the data follow asymmetric distribution. A similar network topology is followed during model development and training. For the uncertainty quantification, ten ensembles were trained in a 67-33% train test split and their outputs were aggregated according to equation (4.27) and (4.28) for the final bound estimation. In the experiment, the parameters (λ & β) were allowed to vary within $[0.1, 1]$. The model comparison was carried out based on the PICP and MPIW quality metrics. A model that achieves the coverage probability PICP $\geq (1 - \alpha)$ with a minimum prediction width is selected

as the optimal model. Since, DBLUBE is distribution based, it has the tendency to overfit. In the experiment, different regularization's techniques were applied to avoid overfitting. However, early stopping was the only approach that addressed overfitting as well as providing a better coverage. Hence, the DBLUBE models were trained for 100 epochs while the models with QD were trained for 500 epochs. The simulation showed that for data that exhibit a skewed distributions (i.e, forest fire, steel factory energy consumption), QD performed well as shown in Table. 3.1. In the case of the forest fire data, we managed to improve DBLUBE's PICP score (PICP:0.94 $\rightarrow$ 0.952, MPIW: 1.71 $\rightarrow$ 1.81) and steel factory energy consumption (PICP:0.94 $\rightarrow$ 0.96, MPIW: 1.76 $\rightarrow$ 1.92) by increasing the batch size. However, under the same network setting, QD handles non-gaussian distributed data much better than DBLUBE. Nevertheless, tuning the QD's λ parameter for better result takes time and makes automatic learning challenging. In addition to that, the need to apply a positive and negative bias to the upper and lower predicted bounds for stable and interpretable forecast creates a numerical instability. On the other hand, under the same network setting and without increasing the depth and width of the network, DBLUBE handles high frequency data better than QD. It is numerically more stable than QD and it can adaptively balance prediction width versus coverage probability without going through the pain of tuning. More importantly, for the same coverage probability, DBLUBE results in the lowest prediction width as compared to others. The experiments have shown that the DBLUBE can provide an alternative learning approach to bound estimation for deep learning models.

4.6 Conclusion

In this chapter, we proposed a simple distribution based upper lower bound estimation algorithm (DBLUBE). The algorithm can adaptively change the variance of the distribution in order to allow a wider or a smaller sampling areas. Consequently, narrowing and widening the prediction interval commensurate with the needed coverage areas. We demonstrated, the algorithm can achieve the desired quality metrics score for data with complex pattern employing a minimum number of layers and neurons as compared to other alternative interval prediction algorithms. Since, the algorithm is distribution based, inherently it is bound to overfit. We observed that this tendency has its own advantage and disadvantage. On one hand, the predicted upper and lower bounds gravitate towards the true distribution which gives us the minimum prediction interval as compared to other algorithms. On the other hand, although it allowed a good model performance during training, it resulted in a poor performance during validation. Especially, when processing a data with a skewed distribution. As such, early stopping has been used as a mitigation strategy to enhance the model generalization ability as well as provide the desired quality metrics.

For data that exhibit a Gaussian-like distribution, the algorithm offers a simple and stable computation with a convenient tuning that ensures a maximized coverage probability and minimized prediction interval. Its response to asymmetric data distribution can be improved by tuning the second auxiliary hyper-parameter(i.e λ_2) towards a gain that provides a higher priority to coverage over a minimized interval.

Chapter 5

Conclusion and Future Work

Grid modernization and the associated big data evolution will see a wider application of machine and deep learning models in different stages of the infrastructure. In addition to the preexisting ones, the massive integration of renewable energies will introduce extra uncertainty in the stability of the grid. As a result, probabilistic predictive models will become ubiquitous in the energy sector. These models will operate on large volume of data. Furthermore, they are expected to deliver an improved prediction accuracy with high coverage probability. One valid approach that is suggested for enhanced model representation and approximation is to free predictive modeling from the shackles of parametric bounds. The intuition behind this approach can attributed to the belief that a model that is not constrained by the number and form of its parameters would have the greater probability of replicating the data generation model. Consequently, there is a shift in modeling assumption where the structure of the model is not determined prior to observing the data. This data driven, structure-less or non-parametric modeling approach tends to be resource intensive compared to the parametric counterpart. However, it exhibits a greater chance of replicating the dynamics of the data generation system. And yet, accurate model representation is not enough for probabilistic models.

Beyond accuracy, probabilistic models need to posses the ability to quantify uncertainty in a such a way that they should guarantee high coverage for the majority of the forecasted points within a minimum confidence interval. By default, being probabilistic means the model will output multiple forecasts for any given point. However, there is no certainty on the optimality of the confidence interval and its coverage probability. This outcome is attributed to the performance metrics employed during model training. Most models are trained either through maximizing the likelihood of the data (MLE) or minimizing the squared deviation (MSE) from the mean. These quality metrics put high emphasis on tracking the mean trajectory without any consideration on the optimality of the returned confidence interval or its coverage probability. In contrary, a training algorithm that give high priority to prediction interval and coverage probability, will produce a better probabilistic model.

With that in mind, in this thesis we proposed a learning algorithm that focus more on the optimality of interval minimization and coverage maximisation. The proposed bound estimation techniques enables deep learning models to have a probabilistic distribution. The effectiveness of the method was demonstrated on a number of real data sets. The experiments have shown that a model trained on this algorithm delivers the desired coverage probability within a minimum prediction interval compared to other alternative methods. Although, the algorithm was successful in training MLP type fully connected neural networks, it didn't results in a consistent bound estimation when applied to the LSTM type neural networks. Our hope being, in our next work we will address the issues

so that the algorithm can be fully integrated.

We have also dedicated our time investigating the implementation of Gaussian process models. We believe that with all its limitations, the Gaussian process deserves a fair amount of attention for two simple reasons, Expressiveness and Interpretability. Of course, there are other predictive models that share these qualities or at least follow a balanced approach to achieve it. But, there are non as natural and intuitive as the Gaussian process. The data driven approach, the ability to map complex patterns and provide flexible functions to a wide range of problems gives them a high degree of expressiveness. Their interpretability, however, lies in the kernel functions. kernels give GP models a simplistic approach to conveniently encode prior assumptions and observed relationships among the elements of the given data. These interactions are later quantified and used as a measure of correlation through model parameters that are interpretable.

Currently, there isn't much freedom in utilizing these models to their fullest extent. The computational requirement induced by the size of the data limit their deployment and more importantly what these models could achieve. Perhaps at some point in the near future, the advancement in computing will likely make the preference between numerical algorithms and the requirement that comes along with it irrelevant. Especially, if researches in the areas of high-performance computing to believed then in accordance with the Moore's law, the next few years will see a faster and improved hardware. May be then, the desirability of these models will be judged based on the true merits of a predictive model such as accuracy and coverage probability than computational demand. Even at this age, GPUs and parallel computing have lessened those demands and are making the implementation of these models a reality in few areas. However, until such time Gaussian approximate inference such as the DTC, FITC and SVGP will fill the void in extending the involvement of these models in major areas.

Bibliography

[1] Kumari, Neha and Kumar, Pushp and Sahu, Naresh Chandra,"Do energy consumption and environmental quality enhance subjective wellbeing in G20 countries?," in Environmental Science and Pollution Research Journal, vol. 28, no. 42, pp. 60246 - 60267, 2021.

[2] European Energy Agency, "Energy prosumers in Europe: Citizen participation in the energy transition," in EEA Report Publication, no. 1, 2022.

[3] International Energy Agency,"World Energy Outlook 2022," in a report publication, 2022.

[4] International Energy Agency,"https://www.iea.org/data-and-statistics/data-tools/renewables-data-explorer," in IEA, 2022.

[5] Ang, Tze-Zhang and Salem, Mohamed and Kamarol, Mohamad and Das, Himadry Shekhar and Nazari, Mohammad Alhuyi and Prabaharan, Natarajan,"A comprehensive study of renewable energy sources: classifications, challenges and suggestions," in Elsevier Energy Strategy Reviews Journal, vol. 43, pp. 100939, 2022.

[6] van Treeck, Ruben, Geist, Juergen, Pander, Joachim, Tuhtan, Jeffrey, Wolter, Christian, "Impacts and Risks of Hydropower, " in Springer Novel Developments for Sustainable Hydropower, pp. 41-60, 2022.

[7] Egli, Florian, "Renewable energy investment risk: an investigation of changes over time and the underlying drivers, " in Elsevier Journal, vol. 140, pp. 111428, 2020.

[8] Shahidehpour, M and Fotuhi-Friuzabad, M, "Grid modernization for enhancing the resilience, reliability, economics, sustainability, and security of electricity grid in an uncertain environment, " in Scientia Iranica Journal, vol. 23, no. 6, pp. 2862 - 2873, 2016.

[9] Santos Neto, Agenor S and Reis, Marcio RC and Coimbra, António Paulo and Soares, Julio CV and Calixto, Wesley P, "Measure of customer satisfaction in the residential electricity distribution service using structural equation modeling, " in MDPI Energies Journal, vol. 15, no. 3 pp.746, 2022.

[10] Qadir, Sikandar Abdul and Al-Motairi, Hessah and Tahir, Furqan and Al-Fagih, Luluwa,"Incentives and strategies for financing the renewable energy transition: A review," in Elsevier Energy Reports, vol. 7, pp. 3590-3606, 2021.

[11] Shokry, Mostafa and Awad, Ali Ismail and Abd-Ellah, Mahmoud Khaled and Khalaf, Ashraf AM,"Systematic survey of advanced metering infrastructure security: Vulnerabilities, attacks, countermeasures, and future vision," in Elsevier Future Generation Computer Systems Journal, 2022.

[12] Amin, BM Ruhul and Taghizadeh, Seyedfoad and Rahman, Md Shihanur and Hossain, Md Jahangir and Varadharajan, Vijay and Chen, Zhiyong,"Cyber attacks in smart grid–dynamic impacts, analyses and recommendations," in IET cyber-Physical Systems: Theory & Applications Journal, vol. 5, no. 4, pp. 321-329, 2020.

[13] Staffell, Iain and Pfenninger, Stefan,"The increasing impact of weather on electricity supply and demand," in Elsevier Energy Journal, vol. 145, pp. 65-78, 2018.

[14] Grandon, Tatiana Gonzalez and Schwenzer, Johannes and Steens, Thomas and Breuing, Julia,"Electricity Demand Forecasting with Hybrid Statistical and Machine Learning Algorithms: Case Study of Ukraine," in arXiv Journal, 2023.

[15] Md Saef Ullah Miah and Junaida Sulaiman and Md. Imamul Islam and Md. Masuduzzaman,"Predicting Short Term Energy Demand in Smart Grid: A Deep Learning Approach for Integrating Renewable Energy Sources in Line with SDGs 7, 9, and 13," in arXiv Journal eprint. 2304.03997, primaryClass. cs.LG, 2023.

[16] Martin Van Creveld,"Seeing into the future," in a book a short History of prediction, 2020.

[17] Yuval Noah,"Sapiens: A brief History of Humankind," book, 2014.

[18] Richard McElreath,"Statistical Rethinking: a Bayesian course with examples, " book. 2020.

[19] Oliver DÜrr, Beate Sick, Elvis Murina," Probabilistic Deep Learning: with python,keras and tensorflow probability," book. 2020.

[20] Khosravi, Abbas and Nahavandi, Saeid and Creighton, Doug and Atiya, Amir F,"Lower upper bound estimation method for construction of neural network-based prediction intervals," in IEEE transactions on neural networks, vol. 22, no. 3, pp. 337-346, 2010.

[21] Shepero, Mahmoud and Van Der Meer, Dennis and Munkhammar, Joakim and Widén, Joakim,"Residential probabilistic load forecasting: A method using Gaussian process designed for electric load data," in Elsevier Applied Energy Journal, vol. 218, pp. 159-172, 2018.

[22] Sun, Xiaokui and Ouyang, Zhiyou and Yue, Dong,"Short-term load forecasting based on multivariate linear regression," in IEEE Conference on Energy Internet and Energy System Integration (EI2), pp. 1-5, 2017.

[23] Juberias, G and Yunta, R and Moreno, J Garcia and Mendivil, C,"A new ARIMA model for hourly load forecasting," in IEEE Transmission and Distribution Conference, vol. 1, pp. 314-319, 1999.

[24] Bercu, Sophie and Proïa, Frédéric,"A SARIMAX coupled modelling applied to individual load curves intraday forecasting," in Applied Statistics Journal, vol. 40, no. 6, pp. 1333- 1348

[25] Rothe, Mrs and Wadhwani, Dr AK and Wadhwani, Dr and others, "Short term load forecasting using multi parameter regression," in arXiv Journal, 2009.

[26] Hong, Wei-Chiang,"Electric load forecasting by support vector model," in Elsevier Applied Mathematical Modelling Journal, vol. 33, no. 5, pp. 2444-2454, 2009.

[27] Fan, Shu and Hyndman, Rob J,"Short-term load forecasting based on a semi-parametric additive model," in IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 134-141, 2011.

[28] Hu, Zhongyi and Bao, Yukun and Chiong, Raymond and Xiong, Tao,"Mid-term interval load forecasting using multi-output support vector regression with a memetic algorithm for feature selection," in Elsevier Energy Journal, vol. 84, pp. 419-431, 2015.

[29] Li, Bin and Lu, Mingzhen and Zhang, Yiyi and Huang, Jia,"A weekend load forecasting model based on semi-parametric regression analysis considering weather and load interaction," in MDPI Energies Journal, vol. 12, no. 20, pp.3820, 2019.

[30] Asber, D and Lefebvre, S and Asber, J and Saad, Maarouf and Desbiens,"Non-parametric short-term load forecasting," in Elsevier International Journal of Electrical Power & Energy Systems, vol. 29, no. 8, pp. 630-635, 2007.

[31] Rafiei, Mehdi and Niknam, Taher and Aghaei, Jamshid and Shafie-Khah, Miadreza and Catalão, João PS,"Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," in IEEE Transcations on Smart Grid, vol. 9, no. 6, pp. 6961-6971, 2018.

[32] Wan, Can and Cao, Zhaojing and Lee, Wei-Jen and Song, Yonghua and Ju, Ping,"An Adaptive Ensemble Data Driven Approach for Nonparametric Probabilistic Forecasting of Electricity Load," in IEEE Transactions on Smart Grid, vol. 12, no. 6, pp. 5396-5408, 2021.

[33] Chaouch, Mohamed,"Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," in IEEE Transactions on Smart Grid, vol. 5, no. 1, pp. 411-419, 2013.

[34] Leith, Douglas J and Heidl, Martin and Ringwood, John V,"Gaussian process prior models for electrical load forecasting," in IEEE International Conference on Probabilistic Methods Applied to Power Systems, pp. 112-117, 2004.

[35] Schulz, Eric and Speekenbrink, Maarten and Krause, Andreas,"A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," in Elsevier Mathematical Psychology Journal, vol. 85, pp. 1-16, 2018.

[36] Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach," a Pearson Higher Education book, 4th Edition, 2019.

[37] Simon Rogers,Mark Girolami, " A first course in machine learning," a second edition book, 2017.

[38] Obst, David and De Vilmarest, Joseph and Goude, Yannig, "Adaptive methods for short-term electricity load forecasting during COVID-19 lockdown in France, " in IEEE transactions on power systems Journal, vol. 36, no. 5, pp. 4754-4763, 2021.

[39] Oliva, Junier B and Dubey, Avinava and Wilson, Andrew G and Póczos, Barnabás and Schneider, Jeff and Xing, Eric P, "Bayesian nonparametric kernel-learning," in PMLR artificial intelligence and statistics proceedings, pp. 1078-1086, 2016.

[40] Duvenaud, David and Lloyd, James and Grosse, Roger and Tenenbaum, Joshua and Zoubin, Ghahramani,"Structure discovery in nonparametric regression through compositional kernel search," in PMLR International Conference on Machine Learning, pp. 1166-1174, 2013.

[41] Calandra, Roberto and Peters, Jan and Rasmussen, Carl Edward and Deisenroth, Marc Peter, " Manifold Gaussian processes for regression," in 2016 IEEE International Joint Conference on Neural Networks (IJCNN), pp. 3338-3345, 2016.

[42] Eressa, Muluken Regas and Badis, Hakim and George, Laurent and Grosso, Dorian," Sparse Variational Gaussian Process with Dynamic Kernel for Electricity Demand Forecasting," in 2022 IEEE 7th International Energy Conference (ENERGYCON), pp. 1-6, 2022.

[43] Daniel Foreman-Mackey and Eric Agol and Sivaram Ambikasaran and Ruth Angus,"Fast and Scalable Gaussian Process Modeling with Applications to Astronomical Time Series," in The Astronomical Jouranl, vol. 154, pp. 220, 2017.

[44] TERRY, Nick et CHOE and Youngju,"Splitting Gaussian processes for computationally-efficient regression," in Plos one Journal, vol. 16, no. 8, pp. e0256470, 2021.

[45] Wilson, Andrew Gordon and Knowles, David A and Ghahramani, Zoubin, "Gaussian process regression networks," in arXiv Journal preprint: 1110.4411, 2011.

[46] Avendaño-Valencia, Luis David and Chatzi, Eleni N and Koo, Ki Young and Brownjohn, James MW,"Gaussian process time-series models for structures under operational variability," in Frontiers in Built Environment Journal, vol. 3, pp. 69, 2017.

[47] Wang, Jack and Hertzmann, Aaron and Fleet, David J,"Gaussian Process Dynamical Models," in MIT Press on Advance in Neural Information processing Systems, vol. 18, 2006.

[48] Sun, A. and Wang, Dingbao and Xu, Xianli,"Monthly streamflow forecasting using Gaussian Process Regression," in Journal of Hydrology, vol. 511, pp. 72-81, Apr. 2014.

[49] Sheng, Hanmin and Xiao, Jian and Cheng, Yuhua and Ni, Qiang and Wang, Song,"Short-term solar power forecasting based on weighted Gaussian process regression," in IEEE Transactions on Industrial Electronics, vol. 65, no. 1, pp. 300-308, 2017.

[50] Yan, Juan and Li, Kang and Bai, Er-Wei and Deng, Jing and Foley, Aoife M,"Hybrid probabilistic wind power forecasting using temporally local Gaussian process," in IEEE Transactions on Sustainable Energy, vol. 7, no. 1, pp. 87-95, 2015.

[51] Richardson, Robert R and Osborne, Michael A and Howey, David A,"Gaussian process regression for forecasting battery state of health," in Elsevier Journal of Power Sources, vol. 357, pp. 209-219, 2017.

[52] Wu, Qi and Law, Rob and Xu, Xin, "A sparse Gaussian process regression model for tourism demand forecasting in Hong Kong," in Elsevier Expert Systems with Applications Journal, vol 39, no. 5, pp. 4769-4774, 2012.

[53] Lourenço, João and Santos, Paulo,"Short term load forecasting using Gaussian process models," in Citeseer Instituto de Engenharia de Sistemas e Computadores de Coimbra Journal, 2010.

[54] Hong, Tao and Fan, Shu,"Probabilistic electric load forecasting: A tutorial review," in Elsevier International Journal of Forecasting, vol. 32, no. 3, pp. 914-938, 2016.

[55] Tomohiro Hachino and Hitoshi Takata and Seiji Fukushima and Yasutaka Igarashi, "Short-Term Electric Load Forecasting Using Multiple Gaussian Process Models, " in International Journal of Electrical and Computer Engineering Journal, vol. 8, no. 2, pp. 447-452, 2014.

[56] Rios, Gonzalo, "Transport gaussian processes for regression," in arXiv Journal preprint. 2011.11473, 2020.

[57] Shawe-Taylor, John and Cristianini, Nello and others, "Kernel methods for pattern analysis, " in a Cambridge university press, 2004.

[58] Sun, Shengyang and Zhang, Guodong and Wang, Chaoqi and Zeng, Wenyuan and Li, Jiaman and Grosse, Roger,"Differentiable compositional kernel learning for Gaussian processes," in PMLR International Conference on Machine Learning, pp. 4828-4837, 2018.

[59] Abdessalem, Anis Ben and Dervilis, Nikolaos and Wagg, David J and Worden, Keith, "Automatic kernel selection for gaussian processes regression with approximate bayesian computation and sequential monte carlo, " in Frontiers in Built Environment Journal, vol. 3, pp. 52, 2017.

[60] Tompkins, Anthony and Ramos, Fabio, "Periodic kernel approximation by index set Fourier series features, " in PMLR Uncertainty in Artificial Intelligence proceedings, pp. 486-496, 2020.

[61] Schwaighofer, Anton and Tresp, Volker and Yu, Kai,"Learning Gaussian process kernels via hierarchical Bayes, " in Advances in neural information processing systems Journal, vol. 17, 2004.

[62] Ma, Tong and Huang, Renke and Barajas-Solano, David and Tipireddy, Ramakrishna and Tartakovsky, Alexandre M,"Electric load and power forecasting using ensemble gaussian process regression" in arXiv Journal, 2019.

[63] Li, Mu and Kwok, James Tin-Yau and Lü, Baoliang, "Making large-scale Nyström approximation possible, " in the ICML 27th International Conference on Machine Learning Proceedings, pp. 631, 2010.

[64] Nguyen-Tuong, Duy and Peters, Jan and Seeger, Matthias, "Local Gaussian process regression for real time online model learning, " in Advances in neural information processing systems Journal, vol. 21, 2008.

[65] Rasmussen, Carl and Ghahramani, Zoubin, "Infinite mixtures of Gaussian process experts, " in Advances in neural information processing systems Journal, vol. 14, 2001.

[66] Titsias, Michalis,"Variational learning of inducing variables in sparse Gaussian processes," in PMLR Artificial intelligence and statistics, pp. 567-574, 2009.

[67] Tajnafoi, Gabor, et al,"Variational gaussian process for optimal sensor placement," in Springer Applications of Mathematics Journal, vol. 66, no. 2, pp. 287-317, 2021.

[68] Hensman, James and Matthews, Alexander and Ghahramani, Zoubin,"Scalable Variational Gaussian Process Classification," in PMLR The Eighteenth International Conference on Artificial Intelligence and Statistics, vol. 38, pp. 351-360, 2015.

[69] Peng, Hao and Zhe, Shandian and Zhang, Xiao and Qi, Yuan,"Asynchronous distributed variational Gaussian process for regression" in PMLR International Conference on Machine Learning, pp. 2788-2797, 2017.

[70] Liu, Haitao and Ong, Yew-Soon and Jiang, Xiaomo and Wang, Xiaofang, "Modulating scalable Gaussian processes for expressive statistical learning, " in Elsevier Pattern Recognition Journal, vol. 120, pp. 108121, 2021.

[71] Salimans, Tim and Kingma, Diederik and Welling, Max,"Markov chain monte carlo and variational inference: Bridging the gap," in PMLR International Conference on Machine Learning, pp. 1218-1226, 2015.

[72] d Ghouse, Jaffer H and Chen, Qi and Eslick, John C and Siirola, John D and Grossman, Ignacio E and Miller, David C, "A flexible framework and model library for process simulation, optimization and control, " in Elsevier Computer Aided Chemical Engineering Journal, vol. 44, pp. 938-942, 2018.

[73] Gal, Yarin and van der Wilk, Mark,"Variational inference in sparse Gaussian process regression and latent variable models-a gentle tutorial," in arXiv Journal, 2014.

[74] Singh, Narendra Pratap and Goh, Sim Kuan and Alam, Sameer,"Real-time unstable approach detection using sparse variational gaussian process," in IEEE International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), pp. 1-10, 2020.

[75] TERRY, Nick et CHOE and Youngju, "Splitting Gaussian processes for computationally-efficient regression," in Plos one Journal vol. 16(8), pp. e0256470, 2021.

[76] Abdar, Moloud and Pourpanah, Farhad and Hussain, Sadiq and Rezazadegan, Dana and Liu, Li and Ghavamzadeh, Mohammad and Fieguth, Paul and Cao, Xiaochun and Khosravi, Abbas and Acharya, U Rajendra and others, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges, " in Elsevier Information Fusion Journal, vol. 76, pp. 243-297, 2021.

[77] Zhou, Zhi-Hua, "Machine learning, " in a book Springer Nature, 2021.

[78] Vanting, Nicolai Bo and Ma, Zheng and Jørgensen, Bo Nørregaard, "A scoping review of deep neural networks for electric load forecasting, " in Springer Energy Informatics Journal, vol. 4, pp. 1-13, 2021.

[79] Su, Hongjun and Zhang, Hong, "On stationary periodic kernels, " in the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), pp. 43-46, 2019.

[80] Suzuki, Joe, "Kernel Methods for Machine Learning with Math and Python: 100 Exercises for Building Logic, " in Springer Nature book, 2022.

[81] Gori, Marco and Betti, Alessandro and Melacci, Stefano, "Machine Learning: A constraint-based approach, " in Elsevier book, 2023.

[82] Deisenroth, Marc Peter and Faisal, A Aldo and Ong, Cheng Soon, "Mathematics for machine learning, " in Cambridge University Press, 2020.

[83] Song, Zhao and Woodruff, David and Yu, Zheng and Zhang, Lichen, "Fast sketching of polynomial kernels of polynomial degree, " in PMLR International Conference on Machine Learning, pp. 9812-9823, 2021.

[84] Wenzel, Tizian and Marchetti, Francesco and Perracchione, Emma, "Data-driven kernel designs for optimized greedy schemes: A machine learning perspective, " in arXiv Journal preprint: arXiv:2301.08047, 2023.

[85] Pennington, Jeffrey and Yu, Felix Xinnan X and Kumar, Sanjiv, "Spherical random features for polynomial kernels, " in Advances in neural information processing systems Journal, vol. 28, 2015.

[86] Theodoridis, Sergios, "Machine learning: a Bayesian and optimization perspective, " in Academic press, 2015.

[87] Gönen, Mehmet and Alpaydın, Ethem, "Multiple kernel learning algorithms, " in The Journal of Machine learning Research, vol. 12, pp. 2211-2268, 2011.

[88] George Casella, Roger L. Berger, " Statistical inference, " by Cengag Learning Publisher, 2021.

[89] Beckers, Thomas, "An introduction to gaussian process models, " in arXiv Journal, preprint: arXiv:2102.05497, 2021.

[90] Huang, Wenbing and Zhao, Deli and Sun, Fuchun and Liu, Huaping and Chang, Edward,"Scalable gaussian process regression using deep neural networks," in the Twenty-fourth international joint conference on artificial intelligence, 2015.

[91] Perone, Christian S and Silveira, Roberto Pereira and Paula, Thomas, "L2M: Practical posterior Laplace approximation with optimization-driven second moment estimation, " in arXiv Journal, preprint: 2107.04695, 2021.

[92] Quiroz, Matias and Nott, David J and Kohn, Robert, "Gaussian Variational Approximations for High-dimensional State Space Models, " in International Society for Bayesian Analysis: Bayesian Analysis Journal, vol. 1, no. 1, pp. 1-28, 2022.

[93] Cheng, Ching-An and Boots, Byron,"Variational inference for Gaussian process models with linear complexity," in arXiv Journal, 2017.

[94] Tölle, Malte and Laves, Max-Heinrich and Schlaefer, Alexander, "A mean-field variational inference approach to deep image prior for inverse problems in medical imaging, " in PMLR Medical Imaging with Deep Learning proceeding, pp. 745-760, 2021.

[95] Nguyen, Trung and Bonilla, Edwin, "Efficient variational inference for Gaussian process regression networks, " in PMLR Artificial Intelligence and Statistics proceedings, pp. 472-480, 2013.

[96] Ong, Victor M-H and Nott, David J and Smith, Michael S, "Gaussian variational approximation with a factor covariance structure, " in Journal of Computational and Graphical Statistics, vol. 27, no. 3, pp. 465-478, 2018.

[97] Lin, Wu and Khan, Mohammad Emtiyaz and Schmidt, Mark, "Fast and simple natural-gradient variational inference with mixture of exponential-family approximations, " in PMLR International Conference on Machine Learning, pp. 3992-4002, 2019.

[98] Hensman, James and Rattray, Magnus and Lawrence, Neil, "Fast variational inference in the conjugate exponential family, " in Advances in neural information processing systems Journal, vol. 25, 2012.

[99] Bauer, Matthias and van der Wilk, Mark and Rasmussen, Carl Edward, "Understanding probabilistic sparse Gaussian process approximations, " in Advances in neural information processing systems Journal, vol. 29, 2016.

[100] Bijl, Hildo and van Wingerden, Jan-Willem and Schön, Thomas B and Verhaegen, Michel, "Online sparse Gaussian process regression using FITC and PITC approximations, " in Elsevier IFAC Journal, vol. 48, no. 28, pp. 703-708, 2015.

[101] Quinonero-Candela, Joaquin and Rasmussen, Carl Edward and Williams, Christopher KI, "Approximation methods for Gaussian process regression, " in MIT lare-scale kernel machines, pp. 203-223, 2007.

[102] Snelson, Edward and Ghahramani, Zoubin,"Sparse Gaussian processes using pseudo-inputs," in Citeseer Advances in neural information processing systems Journal, vol. 18, pp. 1257, 2006.

[103] Ameli, Siavash and Shadden, Shawn C, "A singular Woodbury and pseudo-determinant matrix identities and application to Gaussian process regression, " in Elsevier Applied Mathematics and Computation Journal, vol. 452, pp. 128032, 2023.

[104] Valentin Arkov, "Uncertainty Estimation in Machine Learning, " in arXiv Journal, preprint: arXiv:2206.01749, 2022.

[105] Gogolashvili, Davit and Kozyrskiy, Bogdan and Filippone, Maurizio, "Locally Smoothed Gaussian Process Regression, " in Elsevier Procedia Computer Science Journal, vol. 207, pp. 2717-2726, 2022.

[106] Williams, Christopher, "Computing with infinite networks, " in Advances in neural information processing systems Journal, Vol. 9, 1996.

[107] Lee, Jaehoon and Bahri, Yasaman and Novak, Roman and Schoenholz, Samuel S and Pennington, Jeffrey and Sohl-Dickstein, Jascha, "Deep neural networks as gaussian processes, " in arXiv Journal, preprint: arXiv:1711.00165, 2017.

[108] Kružić, Stanko and Musić, Josip and Kamnik, Roman and Papić, Vladan, "End-Effector Force and Joint Torque Estimation of a 7-DoF Robotic Manipulator Using Deep Learning, " in MDPI Electronics Journal, vol. 10, no. 23, pp. 2963, 2021.

[109] Combalia, M and Hueto, Ferran and Puig, S and Malvehy, J and Vilaplana, Verónica, "Uncertainty estimation in deep neural networks for dermoscopic image classification, ' in IEEE 2020 CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3211-3220, 2020.

[110] Gal, Yarin and Ghahramani, Zoubin, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning, " in PMLR international conference on machine learning, pp. 1050-1059, 2016.

[111] Siddique, Talha and Mahmud, Md Shaad and Keesee, Amy M and Ngwira, Chigomezyo M and Connor, Hyunju, "A Survey of Uncertainty Quantification in Machine Learning for Space Weather Prediction, " in GeoSciences Journal, vol. 12, no. 1, pp. 27, 2022.

[112] Jiang, Weiwei and Luo, Jiayun, "Graph neural network for traffic forecasting: A survey, " in arXiv Journal, preprint. arXiv:2101.11174, 2021.

[113] Cheng, Hsu-Yung and Kuo, Ping-Huan and Shen, Yamin and Huang, Chiou-Jye, "Deep Convolutional Neural Network Model for Short-Term Electricity Price Forecasting, " in arXiv Journal, preprint. arXiv:2003.07202, 2020.

[114] Hiransha, Ma and Gopalakrishnan, E Ab and Menon, Vijay Krishna and Soman, KP, "NSE stock market prediction using deep-learning models, " in Elsevier computer science Journal, vol. 132, pp. 1351-1362, 2018.

[115] Patel, Kinjal and Waslander, Steven, "Accurate Prediction and Uncertainty Estimation using Decoupled Prediction Interval Networks," in arXiv Journal, preprint: arXiv:2202.09664, 2022.

[116] Lai, Yuandu and Shi, Yucheng and Han, Yahong and Shao, Yunfeng and Qi, Meiyu and Li, Bingshuai, "Exploring Uncertainty in Deep Learning for Construction of Prediction Intervals, " in arXiv Journal, preprint. 2104.12953, 2021.

[117] Kivaranovic, Danijel and Johnson, Kory D and Leeb, Hannes, "Adaptive, distribution-free prediction intervals for deep neural networks, " in arXiv Journal, preprint. arXiv:1905.10634, 2019.

[118] Pearce, Tim and Brintrup, Alexandra and Zaki, Mohamed and Neely, Andy, "High-quality prediction intervals for deep learning: A distribution-free, ensembled approach," in PMLR International conference on machine learning proceedings, pp. 4075-4084, 2018.

[119] Pearce, Tim and Anastassacos, Nicolas and Zaki, Mohamed and Neely, Andy, "Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning, " in arXiv Journal, preprint. arXiv:1805.11324, 2018.

[120] Hüllermeier, Eyke and Waegeman, Willem, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, " in Springer Machine Learning Journal, vol. 110, no. 3, pp. 457-506, 2021.

[121] Dolezal, James M and Srisuwananukorn, Andrew and Karpeyev, Dmitry and Ramesh, Siddhi and Kochanny, Sara and Cody, Brittany and Mansfield, Aaron S and Rakshit, Sagar and Bansal, Radhika and Bois, Melanie C and others, "Uncertainty-informed deep learning models enable high-confidence predictions

for digital histopathology, " in Nature communications Journal, vol. 13, no. 1, pp. 6572, 2022.

[122] Graves, Alex, "Practical variational inference for neural networks, " in Advances in neural information processing systems Journal, vol. 24, 2011.

[123] Lakshminarayanan, Balaji and Pritzel, Alexander and Blundell, Charles, "Simple and scalable predictive uncertainty estimation using deep ensembles," in Advances in neural information processing systems Journal, vol. 30, 2017.

[124] Yan, Lian and Verbel, David and Saidi, Olivier, "Predicting prostate cancer recurrence via maximizing the concordance index, " in the tenth ACM SIGKDD international conference on Knowledge discovery and data mining proceedings, pp. 479-485, 2004.

[125] Gross, George and Galiana, Francisco D,"Short-term load forecasting," in IEEE Journal, vol. 75, no. 12, pp. 1558-1573, 1987.

[126] International Energy Agency, "World Energy Outlook 2021," in International Energy Agency Magazine, 2021.

[127] Battaglini, Antonella and Komendantova, Nadejda and Brtnik, Patricia and Patt, Anthony, "Perception of barriers for expansion of electricity grids in the European Union," in Elsevier Energy Policy Journal vol.47, pp. 254-259, 2021.

[128] Rivera, Rodrigo and Burnaev, Evgeny, "Forecasting of commercial sales with large scale Gaussian Processes," in 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 625-634, 2017

[129] Wang, Ye and Ocampo-Martínez, Carlos and Puig, Vicenç and Quevedo, Joseba, "Gaussian-process-based demand forecasting for predictive control of drinking water networks," in Springer International Conference on Critical Information Infrastructures Security, PP. 69-80, 2014.

[130] Alamaniotis, Miltiadis and Chatzidakis, Stylianos and Tsoukalas, Lefteri H,"Monthly load forecasting using kernel based Gaussian process regression," in IET Journal, 2014.

[131] Yan, Bin and Li, Xiwang and Shi, Wenbo and Zhang, Xuan and Malkawi, Ali,"Forecasting building energy demand under uncertainty using gaussian process regression: Feature selection, baseline prediction, parametric analysis and a web-based tool," in 15th IBPSA Conference, pp. 7-9, 2017.

[132] Caywood, Matthew S and Roberts, Daniel M and Colombe, Jeffrey B and Greenwald, Hal S and Weiland, Monica Z,"Gaussian process regression for predictive but interpretable machine learning models: An example of predicting mental workload across tasks," in Frontiers in Human Neuroscience Journal, vol. 10, pp. 647, 2017.

[133] Liu, Xiuming and Zachariah, Dave and Ngai, Edith CH,"Approximate Gaussian Process Regression and Performance Analysis Using Composite Likelihood," in IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6, 2020.

[134] Chen, Yize and Yang, Weiwei and Zhang, Baosen,"Using mobility for electrical load forecasting during the COVID-19 pandemic," in arXiv Journal, 2020.

[135] Tudose, Andrei M and Picioroaga, Irina I and Sidea, Dorian O and Bulac, Constantin and Boicea, Valentin A,"Short-term load forecasting using convolutional neural networks in COVID-19 context: The romanian case study," in MDPI Energies Journal, vol. 14, no. 13, pp. 4046, 2021.

[136] C. E. Rasmussen and C. K. I. Williams,"Gaussian Processes for Machine Learning," in MIT Press Book, 2006.

[137] Titsias, Michalis K,"Variational model selection for sparse Gaussian process regression," a University of Manchester Report, 2009.

[138] Hobbs, Benjamin F and Jitprapaikulsarn, Suradet and Konda, Sreenivas and Chankong, Vira and Loparo, Kenneth A and Maratukulam, Dominic J,"Analysis of the value for unit commitment of improved load forecasts," in IEEE Transactions on Power Systems, vol. 14, no. 4, pp. 1342-1348, 1999.

[139] Limet, Sébastien and Smari, Waleed W and Spalazzi, Luca, " High-performance computing: to boldly go where no human has gone before, " in Concurrency and Computation: Practice and Experience Journal, vol. 27, no. 13, pp. 3145-3165, 2015.

Appendix A

Lets assume that two random vectors $y(x)$ and $f_*(x_*)$ of length n and $n*$ are jointly distributed according to

$$p\left(\begin{bmatrix} f_*(x_*) \\ y(x) \end{bmatrix}\right) \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_*) \\ \mu(x) \end{bmatrix}, \begin{bmatrix} \Sigma_{x_*x_*} & \Sigma_{xx_*} \\ \Sigma_{xx_*}^T & \Sigma_{xx} + \sigma^2 I_d \end{bmatrix}\right) \tag{5.1}$$

For the sake of readability the random variables $f_*(x_*), y(x)$, and the mean functions $\mu(x_*), \mu(x)$ will be referenced as f_*, y and μ_*, μ respectively, ignoring their respective points of observation x and x_*. Now from the multivariate gaussian distribution property, the marginal distribution of y and f_* is given by

$$\begin{aligned} p(y \sim \mathcal{N}(\mu, \Sigma_{xx} + \sigma^2 I_d) \\ p(f_* \sim \mathcal{N}(\mu_*, \Sigma_{x_*x_*}) \end{aligned} \tag{5.2}$$

From the Bayesian inference, the conditional distribution of $p(f_*(x_*) \mid y(x))$

$$p(f_* \mid y) = \frac{p(f_*, y)}{p(y)} \tag{5.3}$$

The marginal distribution of y and its joint distribution with f_* are given as

$$p(y) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{xx} + \sigma^2 I_d|^{1/2}} exp\left(-\tfrac{1}{2}\left(y - \mu\right)^T \left(\Sigma_{xx} + \sigma^2 I_d\right)^{-1} \left(y - \mu\right)\right) \tag{5.4}$$

$$\begin{aligned} p(f_*, y) &= \frac{1}{(2\pi)^{\frac{n+n_*}{2}} |\Sigma|^{1/2}} exp\left(-\tfrac{1}{2}\left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right)^T \Sigma^{-1} \left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right)\right) \\ where\ \Sigma^{-1} &= \begin{bmatrix} \Sigma_{x_*x_*} & \Sigma_{xx_*} \\ \Sigma_{xx_*}^T & \Sigma_{xx} + \sigma^2 I_d \end{bmatrix}^{-1} = \begin{bmatrix} \Sigma_a & \Sigma_b \\ \Sigma_c & \Sigma_d \end{bmatrix},\ |\Sigma| = Det\left(\begin{bmatrix} \Sigma_{x_*x_*} & \Sigma_{xx_*} \\ \Sigma_{xx_*}^T & \Sigma_{xx} + \sigma^2 I_d \end{bmatrix}\right) \end{aligned} \tag{5.5}$$

Applying the matrix inversion lemma will make the conditional distribution estimation much simpler. Hence, the matrix inversion lemma states that if a matrix k is symmetric and positive definite, then its inverse can be given in terms of the Schur complement as

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} (\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} & -(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} & \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1} \end{bmatrix} \tag{5.6}$$

Substituting equation (5.4) and (5.5) into equation (5.3)

$$\begin{aligned} p(f_* \mid y) &= \frac{p(f_*, y)}{p(y)} \\ &= \frac{\frac{1}{\sqrt{(2\pi)^{n+n_*}|\Sigma|}} exp\left(-\tfrac{1}{2}\left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right)^T \Sigma^{-1} \left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right)\right)}{\frac{1}{\sqrt{(2\pi)^n |\Sigma_{xx} + \sigma^2 I_d|}} exp\left(-\tfrac{1}{2}\left(y - \mu\right)^T \left(\Sigma_{xx} + \sigma^2 I_d\right)^{-1} \left(y - \mu\right)\right)} \end{aligned} \tag{5.7}$$

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{xx} + \sigma^2 I_d|}{|\Sigma|}} exp\left(-\tfrac{1}{2}\left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right)^T \Sigma^{-1} \left(\begin{bmatrix} f_* \\ y \end{bmatrix} - \begin{bmatrix} \mu_* \\ \mu \end{bmatrix}\right) + \tfrac{1}{2}\left(y - \mu\right)^T \left(\Sigma_{xx} + \sigma^2 I_d\right)^{-1} \left(y - \mu\right)\right) \tag{5.8}$$

Substituting $\Sigma^{-1} = \begin{bmatrix} \Sigma_a & \Sigma_b \\ \Sigma_c & \Sigma_d \end{bmatrix}$, and expanding equation (5.8), note that $\Sigma_{12} = \Sigma 21$ and $\Sigma_{22} = \Sigma_{xx} + \sigma^2 I_d$

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(-\frac{1}{2}\left(\begin{bmatrix} f_* - \mu_* \\ y - \mu \end{bmatrix}\right)^T \begin{bmatrix} \Sigma_a & \Sigma_b \\ \Sigma_d & \Sigma_d \end{bmatrix} \left(\begin{bmatrix} f_* - \mu_* \\ y - \mu \end{bmatrix}\right) + \frac{1}{2}\left(y-\mu\right)^T \Sigma_{22}^{-1}\left(y-\mu\right)\right)$$

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(\begin{array}{c} -\frac{1}{2}\left((f_*-\mu_*)^T\Sigma_a(f_*-\mu_*) + 2(f_*-\mu_*)^T\Sigma_b(y-\mu) + (y-\mu)^T\Sigma_d(y-\mu)\right) \\ +\frac{1}{2}\left(y-\mu\right)^T \Sigma_{22}^{-1}\left(y-\mu\right) \end{array}\right) \tag{5.9}$$

Substituting the values of $\Sigma_a, \Sigma_b, \Sigma_d$ from equation (5.6) into equation (5.9)

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(\begin{array}{c} -\frac{1}{2}\left((f_*-\mu_*)^T\Sigma_a(f_*-\mu_*) + 2(f_*-\mu_*)^T\Sigma_b(y-\mu) + (y-\mu)^T\Sigma_d(y-\mu)\right) \\ +\frac{1}{2}\left(y-\mu\right)^T \Sigma_{22}^{-1}\left(y-\mu\right) \end{array}\right)$$

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(\begin{array}{c} -\frac{1}{2}\left(\begin{array}{c} (f_*-\mu_*)^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}(f_*-\mu_*) \\ -2(f_*-\mu_*)^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1}(y-\mu) \\ +(y-\mu)^T(\Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1})(y-\mu) \end{array}\right) \\ +\frac{1}{2}\left(y-\mu\right)^T \Sigma_{22}^{-1}\left(y-\mu\right) \end{array}\right)$$

eliminating like terms

$$= \frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(-\frac{1}{2}\left(\begin{array}{c} (f_*-\mu_*)^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}(f_*-\mu_*) \\ -2(f_*-\mu_*)^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1}(y-\mu) \\ +(y-\mu)^T(\Sigma_{22}^{-1}\Sigma_{21}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}\Sigma_{12}\Sigma_{22}^{-1})(y-\mu) \end{array}\right)\right)$$

rearranging terms

$$\frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(-\frac{1}{2}\left((f_*-\mu_*) - \Sigma_{21}\Sigma_{22}^{-1}(y-\mu))^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}((f_*-\mu_*) - \Sigma_{21}\Sigma_{22}^{-1}(y-\mu))\right)\right)$$

$$\frac{1}{\sqrt{(2\pi)^{n_*}}} \sqrt{\frac{|\Sigma_{22}|}{|\Sigma|}} exp\left(-\frac{1}{2}\left((f_* - (\mu_* + \Sigma_{21}\Sigma_{22}^{-1}(y-\mu)))^T(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1}(f_* - (\mu_* + \Sigma_{21}\Sigma_{22}^{-1}(y-\mu)))\right)\right) \tag{5.10}$$

Which is a probability density for a multivariate distribution with mean $(\mu_* + \Sigma_{21}\Sigma_{22}^{-1}(y-\mu))$ and covariance $(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$. Hence, the conditional distribution of f_* given y is estimated as

$$\begin{aligned} p(f_* \mid y) = \frac{p(f_*, y)}{p(y)} &\sim \mathcal{N}(m(x), \Sigma(x)) \\ &\sim \mathcal{N}(\mu_* + \Sigma_{21}\Sigma_{22}^{-1}(y-\mu), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \end{aligned} \tag{5.11}$$