Advances in computational Bayesian statistics and the approximation of Gibbs measures

James Ridgway

Université Paris-Dauphine
École Doctorale de Dauphine
Centre de Recherche en Mathématiques de la Décision

Thèse presentée par:

James L.P. Ridgway

Pour obtenir le grade de:

Docteur en Mathematiques Appliquées
Spécialité: Statistiques

ADVANCES IN COMPUTATIONAL BAYESIAN STATISTICS AND THE APPROXIMATION OF GIBBS MEASURES

Jury composé de:

M. Christophe ANDRIEU	Bristol University	Rapporteur
M. Olivier CATONI	CNRS CREST-ENSAE	Rapporteur
M. Nicolas CHOPIN	CREST-ENSAE	Directeur de Thèse
M. Randal DOUC	Telecom SudParis	Examinateur
M. Erwan LE PENNEC	École Polytechnique	Examinateur
M. Christian ROBERT	Université Paris Dauphine	Examinateur

Remerciements

Je voudrais tout d'abord remercier vivement mon directeur de thèse, Nicolas Chopin, pour son engagement, sa disponibilité et son enthousiasme contagieux lors de nos discussions scientifiques. Pour toutes ces raisons, travailler sous sa direction a été une exprience très agréable et enrichissante.

J'adresse également mes remerciements à Randal Douc, Erwan Le Pennec et Christian Robert d'avoir accepté de participer à mon jury de thèse ainsi qu'à Christophe Andrieu et Olivier Catoni pour avoir accepté de la rapporter et pour leur lecture attentive et leurs commentaires constructifs.

Je remercie les doctorants ayant séjourné au bureau E28: Adélaïde, Edwin, JB et Mathieu ainsi que ceux du CREST et de Dauphine: Clara, Marco, Medhi, Pierre et Vincent qui ont contribué à rendre ces trois années agréables.

Ma gratitude va également au CREST pour son environnement de travail très stimulant. Les chercheurs, notamment Pierre Alquier, Arnak Dalalyan et Judith Rousseau, ont toujours t disponibles et m'ont beaucoup appris sur le plan scientifique. Merci aussi au CREST d'avoir mis la salle café à côté de notre bureau, décuplant ainsi notre productivité!

Ma reconnaissance va également à mes parents et à Vicky pour leur soutien et les dimanches soirs passés en leur compagnie. Je pense aussi à mes amis (particulièrement Zago, Seb, Yann et Khalid) grâce auxquels j'ai passé de bons moments tout au long de cette aventure. Je n'oublie pas non plus les joueurs RC Val de Bièvre pour toutes les victoires qui ont rythmé ces trois dernières années. Enfin, je dédie ce mémoire de thèse à Céline dont le soutien ne m'a jamais fait défaut.

Contents

Resumé

Ce mémoire de thèse traite de plusieurs méthodes de calcul d'estimateur en statistiques bayésiennes. Le premier chapitre consiste en une brève introduction des thèmatiques abordées. Nous en donnons ici une première vue d'ensemble.

Plusieurs approches d'estimation seront considérées dans ce manuscrit. D'abord en estimation nous considérerons une approche standard dans le paradigme bayesien en utilisant des estimateurs sous la forme d'intégrales par rapport à des lois *a posteriori*. Dans un deuxième temps nous relacherons les hypothèses faites dans la phase de modélisation. Nous nous intéresserons alors à l'étude d'estimateurs répliquant les propriétés statistiques du minimiseur du risque de classification ou de *ranking* théorique et ceci sans modélisation du processus génératif des données.

Dans les deux approches, et ce malgré leur dissemblance, le calcul numérique des estimateurs nécessite celui d'intégrales de grande dimension. La plus grande partie de cette thèse est consacrée au développement de telles méthodes dans quelques contextes spécifiques.

Nous diviserons les algorithmes en deux grandes classes. D'abord les algorithmes de Monte Carlo basés sur la génération de variables aléatoires dans l'épigraphe de l'intégrande. Ces derniers permettent le calcul d'intégrales dans la mesure ou nous arrivons à générer efficacement de tels points. Le chapitre 2 compare certaines des méthodes couramment utilisées pour l'estimation bayésienne de modèles probit. Nous y donnons des recommandations sur la méthodologie à adopter. Dans le chapitre 3 nous développerons un algorithme permettant le calcul de probabilités gaussiennes de rectangles de grandes dimensions. Ce chapitre traite un problème plus géneral que celui de l'estimation, cependant, nous pouvons lier la problématique à celle de l'évaluation de certaines vraisemblances.

La deuxième classe d'algorithme que nous considérons consiste en l'approximation de distributions par des distributions dont les moments peuvent être calculés ex-

plicitement. Il s'agira dans ce cas de définir une métrique et de trouver les approximations les plus proches dans une classe donnée. Dans le chapitre 4 nous étudierons les proprietés des mesures de Gibbs pour répliquer les propriétés du minimiseur d'un risque de *ranking*. Nous développerons également des méthodes d'approximation pour ces lois. Dans le chapitre 5 nous nous intéresserons plus spécifiquement à une manière d'approcher les distributions et nous étudierons les propriétés theoriques des approximations elles-mêmes, i.e. leurs capacités à répliquer des propriétés du minimiseur du risque.

Une description plus détaillée de chaque chapitre est donnée en fin d'introduction.

1 Introduction

In this thesis we study some computational aspects of Bayesian statistics, as well as Gibbs posteriors. We describe both statistical approaches in the first two sections. We then give a brief overview of computational aspects linked to the implemention of those estimators in Section 1.3 of this chapter. This chapter is devoted to discussing the different issues that will arise in the manuscript; detailed accounts will be found in the following chapters.

1.1 Bayesian statistics

Statistical analysis starts with a collection of probability distributions P_θ indexed by a parameter $\theta \in \Theta$, where Θ is an arbitrary set. In this thesis most examples will be taken from parametric statistics, with $\Theta \subset \mathbb{R}^d$. The statistician then confronts the model to some observations on the probability space $(\mathcal{X}, \mathcal{A}, \{P_\theta, \theta \in \Theta\})$.

The goal of this thesis is not the choice of the collection of probability distribution but rather, given the model and the data, to discuss ways of improving the computation of estimators of θ.

We describe in the following the Bayesian paradigm that will be used in this manuscript. In particular we discuss the minimization of the integrated posterior risk. As we will see, this criterion leads to computational difficulties because it requires the evaluation of high dimensional integrals.

In Section 1.2, we will take another approach: we will not suppose a specific probability model but rather minimize an empirical risk over classes of classifiers and give theoretical guaranties for such an approach.

In all the following we suppose the model to be dominated by some measure and we define the likelihood as the probability density of the observation given the parameter.

Definition 1.1.1 *We call likelihood the probability density of the collection* $(X_{1:n}) \in \mathcal{X}^n$ *conditioned on* θ*. We denote it by* $\mathcal{L}(X_{1:n}|\theta)$*.*

We give two examples of models that will be used in the rest of the thesis. Those models will be used in the introduction as examples and will be further studied in the different chapters of this thesis.

Example 1.1.1 *Probit. This is a special case of generalized linear models where the probability of a binary random variable is expressed as a linear combination of some covariates. Conditionally on* θ *and a vector of deterministic covariates* x *the model can be expressed in a hierarchical form:*

$$\begin{aligned} Y_i &= \mathbb{1}_{Z_i>0}, \\ Z_i|\beta &\sim \mathcal{N}(x_i\beta, 1). \end{aligned}$$

Each observation is sampled independently from this model. The likelihood is given by

$$\mathcal{L}(y_{1:n}|\theta) = \prod_{i=1}^{n} \left(\Phi(x_i\theta)\right)^{y_i} \left(1 - \Phi(x_i\theta)\right)^{1-y_i}.$$

This model will be studied in details in Chapter 2. In Chapter 3 we give an example of a model where we allow correlations between the $(Z_i)_i$ and a way to compute the likelihood.

Example 1.1.2 *State space model (SSM). SSM is another model that we will use in this thesis (see Chapter 6). This is a time series model with an unobserved Markov process* $(x_t)_{t\geq 0}$ *with transition density*

$$x_0 \sim g_0(.;\theta); \qquad x_t|x_{1:t-1} \sim g_t(.|x_{t-1};\theta)$$

and an observation y_t *at each time* t *depending on the current value of the chain,*

$$y_t \sim f_t(.|x_t;\theta).$$

The likelihood of the observations up to time T *is given by*

$$\mathcal{L}(y_{1:T};\theta) = \int \prod_{t=1}^{T} f(y_t|x_t;\theta) g_t(x_t|x_{t-1};\theta) \mu_0(x_0) \mathrm{d}x_{0:T}.$$

The fact that the likelihood takes the form of an integral with respect to a Markov process leads to computational issues. Those will be discussed in more details in Chapter 6.

The main idea of the Bayesian paradigm is to endow Θ with the structure of a probability space $(\Theta, \mathcal{B}, \pi)$ where the probability measure π is referred to as the prior. We do not discuss how to choose π; discussion of this subject can be found in Robert [2007].

Once we have defined the prior, we can use the likelihood as a conditional probability and using Bayes identity one can define the posterior distribution:

Definition 1.1.2 *The posterior distribution is the distribution of the parameter given the observations,*

$$\pi(\mathrm{d}\theta|\mathcal{D}) = \frac{\mathcal{L}(x_{1:n}|\theta)\pi(\mathrm{d}\theta)}{\int_\Theta \mathcal{L}(x_{1:n}|\theta')\pi(\mathrm{d}\theta')}.$$

We call the integral $m_\pi(x_{1:n}) := \int_\Theta \mathcal{L}(x_{1:n}|\theta')\pi(\mathrm{d}\theta')$ *the marginal likelihood.*

The posterior is the distribution of the parameter given a fixed observed sample. We derive in the following section criteria to choose an estimator using this distribution.

1.1.1 Risk minimizer

Given a model $\{P_\theta, \theta \in \Theta\}$, the data and a prior $\pi(\mathrm{d}\theta)$, with support Θ, the goal of the statistician is to take a decision δ minimizing a risk. Several types of risk can be considered and lead to different estimators. We now give examples that are used in this manuscript.

Once we have defined the risk one cannot simply minimize over value of ℓ because of the unknown θ. To deal with this issue within the Bayesian paradigm the statistician defines the posterior risk.

Definition 1.1.3 *The Posterior risk is the loss function integrated with respect to the posterior distribution:*

$$\rho(\pi, \delta|\mathcal{D}) = \int \ell(\theta, \delta)\pi(\mathrm{d}\theta|\mathcal{D}).$$

A Bayesian estimator is a minimizer of the posterior risk. The most usual loss is the quadratic loss $\ell(\delta, \theta) = \|\theta - \delta\|_2^2$; direct computation yields $\delta^\pi = \mathbb{E}_\pi(\theta|\mathcal{D})$. An other common loss is the following 0-1 loss,

$$\ell(\theta, \delta) = \mathbb{1}_{\Theta_1}(\theta)\mathbb{1}_{\delta=1} + \mathbb{1}_{\Theta_2}(\theta)\mathbb{1}_{\delta=0}.$$

Direct computations yield:

$$\delta^\pi = \begin{cases} 1 \text{ if } \mathbb{P}_\pi(\theta \in \Theta_1|\mathcal{D}) > \mathbb{P}_\pi(\theta \in \Theta_2|\mathcal{D}) \\ 0 \text{ otherwise} \end{cases}$$

In the light of those examples we see the main computational issue appear. Those two examples show that the paradigm leads to non trivial integration problems. In both case one wants to compute:

$$\frac{\int_\Theta h(\theta)\pi(\mathrm{d}\theta)\mathcal{L}(\theta|\mathcal{D})}{\int_\Theta \pi(\mathrm{d}\theta)\mathcal{L}(\theta|\mathcal{D})},$$

for some function h; i.e. an expectation with respect to a distribution that is known only up to the normalizing constant $m_\pi(\mathcal{D}) = \int_\Theta \pi(\mathrm{d}\theta)\mathcal{L}(\theta|\mathcal{D})$.

1.1.2 Model choice

We can encode model uncertainty by defining the parameter space $\mathbf{\Theta} = \prod_j \Theta_j \times \{j\}$ where $\{j\}$ indexes the model and Θ_j is the parameter space of models j. The standard approach in the Bayesian paradigm is to endow the space of model indexes with a prior distribution and to treat it as an additional parameter. This is easily encapsulated in the framework described in the previous section.

Most of the time the likelihood itself will not be tractable in this case. We can still solve the problem using specially tailored MCMC algorithms [Green, 1995]. In Chapter 2 we will propose an alternative to this for the case of covariate uncertainty for probit models.

We have defined the normalizing constant of the posterior in Definition 1.1.2 as $m_\pi(x) = \int \pi(\theta)\mathcal{L}(x|\theta)\mathrm{d}\theta$. We write the likelihood of model j, $m_\pi(x|j)$. We assign a prior probability π_j to model j. We end up with an integration problem with respect to the following distribution,

$$\pi(j|\mathcal{D}) = \frac{\pi_j m_\pi(x|j)}{\sum_j \pi_j m_\pi(x|j)}.$$

For covariate selection in a Gaussian linear model one can compute the marginal likelihood provided that the prior is conjugate to the Gaussian family (i.e. a Gaussian distribution). An example of the approach is given in Chopin and Shaeffer [2011]. In general the marginal likelihood is not tractable. A way to get around this issue is to replace it by an estimator (see Chapter 2 for an application to the probit model).

1.2 PAC-Bayesian Bounds

When no clear mathematical model is available to the statistician or that existing models are too costly to evaluate or to build, one may still be able to construct a pseudo-posterior to replicate the behavior of the risk minimizer. We use for

this matter a Gibbs posterior (defined bellow) for the chosen risk. For this posterior we prove nonasymptotic bound. The idea originated in machine learning (Shawe-Taylor and Williamson [1997],McAllester [1998]) as way to bound the theoretical risk in probability by a computable empirical quantity. We follow Catoni [2007] and use the approach to get oracle inequalities on the risk integrated under our given pseudo-posterior.

We observe a sample $(X_1, Y_1), \cdots, (X_n, Y_n)$ taking values in $\mathcal{X} \times \mathcal{Y}$ where the pairs (X_i, Y_i) have the same distribution $\mathbb{P}$.

The statistician defines a set of predictor $\{f_\theta : \mathcal{X} \to \mathcal{Y}\}$. We suppose we also have at our disposal a risk function $R(\theta)$ and its empirical counterpart $r_n(\theta)$.

The approach is summarized for classification in this section. That is we take $\mathcal{Y} = \{-1, 1\}$, $R(\theta) = \mathbb{P}(Y f_\theta(X) \leq 0)$ and $r_n(\theta) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{Y_i f_\theta(X_i) \leq 0}$. In the following we also suppose linear classifiers $f_\theta(x) = 2\mathbb{1}_{<x,\theta> \geq 0} - 1 \in \{-1, 1\}$. More details on the derivation of the estimator and theoretical results are given in Chapter 4 and 5.

Definition 1.2.1 *The Gibbs posterior for an empirical risk $r_n(\theta)$ and a prior π_ξ is given by*

$$\hat{\rho}_\lambda(\mathrm{d}\theta) = \frac{1}{Z_{\lambda,\xi}} \exp\left\{-\lambda r_n(\theta)\right\} \pi_\xi(\mathrm{d}\theta)$$

where ξ is the vector of hyperparameters, and $Z_{\xi,\lambda} = \int_\Theta e^{-\lambda r_n(\theta)} \pi_\xi(\mathrm{d}\theta)$.

Contrarily to the posterior defined in the preceding section we do not derive this probability density from Bayes' formula. The Gibbs posterior defined in definition 1.2.1 is the solution of the following variational problem:

$$-\log Z_{\lambda,\xi} = \inf_{\rho \in \mathcal{M}_+^1} \left\{\lambda \rho(r_n(\theta)) + \mathcal{K}(\rho, \pi_\xi)\right\},$$

where $\mathcal{M}_+^1$ is the set of all probability measures. This is easily deduced from the following equality: for any $\rho \in \mathcal{M}_+^1$

$$-\log Z_{\lambda,\xi} + \mathcal{K}(\rho, \hat{\rho}_\lambda) = \lambda \rho(r_n(\theta)) + \mathcal{K}(\rho, \pi_\xi).$$

Part of this thesis is concerned with proving oracle inequalities for those measures and providing approximations to the Gibbs posterior. We give a quick overview of an approach to derive bounds for the expected risk under a Gibbs posterior. We follow Catoni [2007] and refer the reader to this reference for a complete account and tighter bounds.

We give a simplified proof for the classification loss, to give an idea of the tools used in Chapter 4 and 5. In Chapter 4 we will extend these results to the case of an AUC loss. The loss will be used to propose a method to rank instances from bipartite data.

Using the above relation we derive the following results,

$$\begin{aligned}
&\mathbb{P}\left\{\sup_{\rho\in\mathcal{M}_+^1} \lambda\left(\rho(r_n(\theta)) - \rho(R(\theta))\right) + \mathcal{K}(\rho,\pi_\xi) + \eta \geq 0\right\} \\
&= \mathbb{P}\left\{\log \pi_\xi\left[\exp\left\{-\lambda(r_n(\theta) - R(\theta)) + \eta\right\}\right] \geq 0\right\} \\
&= \mathbb{P}\left\{\pi_\xi\left[\exp\left\{-\lambda(r_n(\theta) - R(\theta)) + \eta\right\}\right] \geq 1\right\} \\
&\leq \mathbb{P}\pi_\xi\left[\exp\left\{-\lambda(r_n(\theta) - R(\theta)) + \eta\right\}\right] \\
&= \pi_\xi\mathbb{P}\left[\exp\left\{-\lambda(r_n(\theta) - R(\theta)) + \eta\right\}\right].
\end{aligned}$$

Since $r_n(\theta)$ is bounded from above, Hoeffding's inequality allows us to write:

$$\pi\mathbb{P}\left[\exp\left\{-\lambda(r_n(\theta) - R(\theta)) + \eta\right\}\right] \leq \pi\left\{\exp\left(\frac{\lambda^2}{2n} + \eta\right)\right\}$$

see Lemma 2.2 an Theorem 2.1 in Boucheron et al. [2013]. A great part of the effort will be put on finding concentration inequality at this step. In Chapter 4 for instance we prove a Bernstein inequality for the AUC risk to this aim.

We put $\eta = -\frac{\lambda^2}{2n} - \log\frac{2}{\epsilon}$ and get $\forall\rho \in \mathcal{M}_+^1$ with probability less than $\epsilon/2$:

$$\lambda\left(\rho(R(\theta)) - \rho(r_n(\theta))\right) + \mathcal{K}(\rho,\pi) - \frac{\lambda^2}{2n} - \log\frac{2}{\epsilon} \geq 0 \tag{1.1}$$

such that with probability at least $1-\epsilon/2$ we can upper bound the theoretical risk under the Gibbs posterior

$$\hat{\rho}_\lambda(R\theta) \leq \inf_{\rho\in\mathcal{M}_+^1}\left\{\rho(r_n(\theta)) + \frac{1}{\lambda}\mathcal{K}(\rho,\pi)\right\} + \frac{\lambda}{2n} + \frac{1}{\lambda}\log\frac{2}{\epsilon}. \tag{1.2}$$

Chapter 4 will be concerned with extensions of the framework to rank data, and with computational tools to compute approximations of the Gibbs posterior.

Notice that this intermediate result comes under the only hypothesis that the data is *iid*. In fact we can even weaken this assumption by taking a weakly dependent sample (see for instance Alquier and Li [2012]).

1.2.1 Minimizing the bound

Another quantity of interest is the normalizing constant. In the framework described previously we can write inequality (1.2) using the dual formulation introduced in the beginning of the section w.p. at least $1-\epsilon$

$$\hat{\rho}(R(\theta)) \leq -\frac{1}{\lambda}\log Z_{\lambda,\xi} + \frac{\lambda}{2n} + \frac{1}{\lambda}\log\frac{2}{\epsilon}.$$

Several authors have proposed to use this bound not only to give an empirical bound on the true risk but also for estimation. That is, choose the hyper-parameter of the prior or an approximation of the model, such that the empirical bound is the tightest. The log-normalizing constant is not always easy to compute, however we show in the following sections some tools to find an unbiased estimator of this integral. In Chapter 5 we show how to replace the optimal measure by the optimal measure in a smaller class of distribution for which the integral is tractable.

1.2.2 PAC Bayesian oracle inequality

We call oracle some estimator based on an unobservable quantity. In most of this thesis it will be given by the minimizer of the theoretical risk. Here we write $\bar{\theta} = \arg\min_{\theta\in\Theta} R(\theta)$. From equation 1.1 we get simultaneously $\forall\rho \in \mathcal{M}_+^1(\Theta)$ with probability $1-\epsilon$,

$$\rho(R(\theta)) \leq \rho(r_n(\theta)) + \frac{1}{\lambda}\mathcal{K}(\rho,\pi_\xi) + \frac{\lambda}{2n} + \frac{1}{\lambda}\log\frac{2}{\epsilon},$$
$$\rho(r_n(\theta)) \leq \rho(R(\theta)) + \frac{1}{\lambda}\mathcal{K}(\rho,\pi_\xi) + \frac{\lambda}{2n} + \frac{1}{\lambda}\log\frac{2}{\epsilon}.$$

Specifying ρ as the Gibbs posterior in the first of the two equations and replacing $\rho(r_n)$ by its upper bound one gets with probability $1-\epsilon$,

$$\hat{\rho}_\lambda(R(\theta)) \leq \inf_{\rho\in\mathcal{M}_1^+}\left\{\rho(R(\theta)) + \frac{2}{\lambda}\mathcal{K}(\rho,\pi_\xi)\right\} + \frac{\lambda}{n} + \frac{2}{\lambda}\log\frac{2}{\epsilon}.$$

The inequality gives rise to a bound on the integrated theoretical risk. To obtain a specific bound and introduce the oracle risk we will introduce an additional assumption.

Definition 1.2.2 *There exists a constant $c > 0$ such that $\forall(\theta,\theta') \in \Theta^2$ with $\|\theta\| = \|\theta'\| = 1$ we have $\mathbb{P}\left(< X,\theta >< X,\theta' >\leq 0\right) \leq c\|\theta-\theta'\|$.*

The assumption implies that for any two given linear classifiers we can control the amount of points that are classified differently. More specifically, suppose that X admits a density, with respect to the $d-1$ spherical measure, upper bounded by B then we can show,

$$\begin{aligned}\mathbb{P}\left\{< X,\theta >< X,\theta' >\leq 0\right\} &\leq \frac{B}{2\pi}\arccos(< \theta,\theta' >)\\ &\leq \frac{B}{2\pi}\sqrt{\frac{5}{2}}\|\theta-\theta'\|\end{aligned}$$

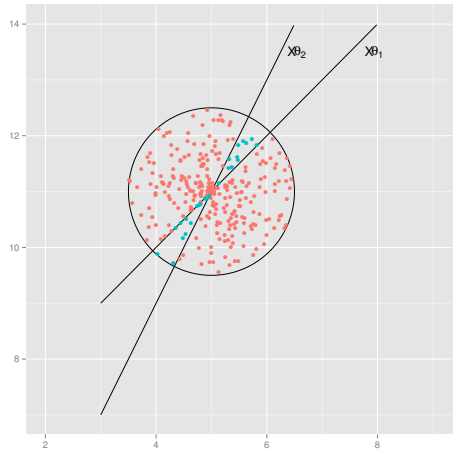


Figure 1.1: Illustration of the assumption

The hypothesis imposes that for any two classifiers there is enough mass in the region where their conclusion differs.

By specifying the prior to be a Gaussian distribution with variance ϑ we get an oracle inequality. That is we take $\pi(\theta) = \prod_{i=1}^{d} \varphi(\theta_i; 0, \vartheta^2)$. Several other prior can be considered (see Chapter 4 for an application to spike and slab to induce sparsity).

Theorem 1.2.1 *Take $\lambda = \sqrt{nd}$ and $\vartheta = \frac{1}{\sqrt{d}}$. Under the assumption of Definition 1.2.2, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have,*

$$\hat{\rho}_\lambda(R(\theta)) \leq R(\bar{\theta}) + \sqrt{\frac{d}{n} \log\left(4ne^2\right)} + \frac{c}{\sqrt{n}} + \sqrt{\frac{d}{4n^3}} + \frac{2\log\left(\frac{2}{\varepsilon}\right)}{\sqrt{nd}}.$$

The proof of this result is provided in Chapter 5. Note that we can also deduce bounds in expectation from bound of Theorem 1.2.1, using the fact that $\int_0^\infty \mathbb{P}\{X > t\}\,\mathrm{d}t = \mathbb{E}(X)$.

In Chapter 4 we specify other types of priors with the aim of dealing with different hypotheses i.e. a spike and slab for sparsity and a Gaussian process prior for non-linear classifier.

The issue with the result is that the estimator is intractable. We will give ways to approximate this measure in the subsequent chapters. In Chapter 5 in particular we give an approximation for which we can obtain oracle inequalities directly.

1.3 Computational aspects

The general idea behind Bayesian computational statistics is easily summed up by Minka [2001a]. "[...] instantiate the things we know, and integrate over the things we don't know, to compute whatever expectation or probability we seek" We will therefore be interested in computing the integral

$$\int_\Theta h(\theta)\pi(\mathrm{d}\theta|\mathcal{D}).$$

We have seen that for a quadratic loss the optimal estimator is the posterior mean. But we can also take interest in other quantities such as moments, the mode, posterior probabilities etc. In all these cases we have to deal with the intractability of the integral with respect to the posterior as well as the intractability of the normalizing constant. We will divide the introduction of the different algorithms in two sections, the first dealing with Monte Carlo , i.e. based on random samples, the second on deterministic variational approximations.

1.3.1 Monte Carlo

Principle of Monte Carlo

Monte Carlo is based on the use of the law of large numbers (LLN) to approximate integrals. We will be interested in computing integrals of the form $I_h := \mathbb{E}_{\mathbb{P}}(h(X))$. The LLN gives us, under the hypothesis that the $(X_i)_{\geq 0}$ are iid $\mathbb{P}$ and the existence of I_h, that:

$$\frac{1}{N}\sum_{i=1}^{N} h(X_i) \xrightarrow[N\to\infty]{\mathbb{P}} I_h.$$

A stronger result is given by the central limit theorem under the hypothesis of finite second order moment,

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} h(X_i) - I_h\right) \xrightarrow[N\to\infty]{\mathcal{L}} \mathcal{N}(0,\sigma^2),$$

with σ^2 the variance of $h(X)$ under $\mathbb{P}$.

We need a way to sample from $\mathbb{P}$ in the most general way possible to apply these results. Several approaches exist for sampling, the two direct ones one may consider are:

- Inverting the c.d.f: for sampling from a distribution with c.d.f F one can sample a uniform $U \sim \mathcal{U}_{[0,1]}$ and compute $Y = F^{-1}(U)$. This requires knowledge of the c.d.f. which we have seen will not happen often in practice,

and the ability to compute its inverse or pseudo-inverse. We give an example below for the truncated Gaussian distribution which we will use repeatedly in Chapter 3.

- Rejection sampling Accept reject (AR) is based on the fact that sampling uniformly in the epigraph of a density leads to points marginally distributed under it. The additional layer here is that we use a proposal distribution g such that uniformly on the support of the target f we have that $f \leq Mg$ for a known constant M. Then AR consists in sampling from the joint distribution $(U, X) \sim M\mathbb{1}_{\left[0 \leq U \leq \frac{f(X)}{Mg(X)}\right]} g(X)$. The proposal g is a degree of freedom for the user and will impact the efficiency of the algorithm. Finally, note that for log concave densities there exists algorithms to automatically construct g, see Chapter 3 of Robert and Casella [2004b] for an example.

In the above we have left one question unanswered: all of the algorithms require the ability to generate $i.i.d$ sequence of uniform distribution. In this thesis we will consider that such an algorithm is given. An account is given in Niederreiter [1978].

We give an example in the following on how to apply the above methodology to a truncated Gaussian (see Chapter 3).

Example 1.3.1 *Truncated Normal distribution*

The aim is to sample $Y \sim \mathcal{N}_{[a,b]}(m, \sigma^2)$. *The c.d.f of a* (m, σ^2) *Gaussian truncated to* $[a, b]$ *is given by:*

$$F(x; m, \sigma, a, b) = \frac{\Phi\left(\frac{x-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)}{\Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)}$$

leading to the algorithm consisting in sampling a uniform distribution U *on* $[0, 1]$ *and taking*

$$Y = m + \sigma\Phi^{-1}\left\{U\left(\Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right)\right) + \Phi\left(\frac{a-m}{\sigma}\right)\right\}$$

This particular sampler will be used extensively in Chapter 3.

An AR algorithm has also been developed for this example [Robert, 1995], using a translated exponential where the parameters are chosen as to maximize the acceptance probability. Chopin [2011a] proposes a more efficient algorithm based on the Ziggurat algorithm.

Importance sampling

Importance sampling is based on the identity

$$\mathbb{E}_{\mathbb{P}}\left(h(X)\right) = \mathbb{E}_{\mathbb{Q}}\left(h(X)\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(X)\right)$$

under the hypothesis that $\mathbb{Q} >> \mathbb{P}$ (measure $\mathbb{Q}$ dominates $\mathbb{P}$).

As previously importance sampling is based on replacing the integration with respect to $\mathbb{Q}(\mathrm{d}x)$ with its empirical counterpart $\mathbb{Q}_n(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_i}(\mathrm{d}x)$ with $X_i \sim \mathbb{Q}(\mathrm{d}x)$ hence the estimator $I_h \approx \frac{1}{M}\sum_{i=1}^{M} w_i h(X_i)$, where we write $w_i(X_i) := \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(X_i)$.

It is often the case that the weights w are known only up to a normalizing constant (in particular in the case of posterior sampling). One can use a normalized estimator. That is we will replace the unknown normalizing constant by the corresponding IS estimator $\frac{1}{M}\sum_{i=1}^{M} w_i(X_i)$. Because this is a convergent estimator bounded away from 0, one can show that the ratio

$$\frac{\sum_{i=1}^{M} w_i(X_i)h(X_i)}{\sum_{i=1}^{M} w_i(X_i)}$$

converges towards the correct integral. One can get results for finite M through concentration inequalities (Cappé et al. [2005] chapter 9).

In the rest of the chapter we suppose that both measures are dominated with respect to a base measure and write $\frac{p}{q}(X)$ the ratio of densities with respect to this measure.

Remark 1.3.1 *Unbiased target*

It is interesting to note that importance sampling remains valid if one replaces the weights with an unbiased estimator. Suppose we have a quantity $\hat{T}_Z(X)$ such that $\mathbb{E}_{\mathbb{S}}\hat{T}_Z(X) = \frac{p}{q}(X)$. In this case we can build an algorithm consisting in sampling $Z_i \sim \mathbb{S}$ and $X_i \sim \mathbb{Q}$ and forming the estimator

$$\hat{I}_h = \frac{1}{N}\sum_{i=1}^{N}\hat{T}_{Z_i}(X_i)h(X_i),$$

one readily sees that integrating with respect to Z yields an IS estimator of I_h.

In the following we give a running example that we use to illustrate the case of unbiased targets. In Chapter 6 we will use such results in the context of state space models extending the algorithm of Chopin et al. [2013b].

Example 1.3.2 *Symmetric Skellam distribution*

The difference of two random variables with Poisson distributions is known as the Skellam distribution. We can write its density as a convolution between a Poisson and a negative Poisson,

$$p(z;\lambda) = \sum_{i=0}^{\infty} \varpi(i+z;\lambda)\varpi(i;\lambda). \tag{1.3}$$

where $\varpi(x;\lambda)$ is the Poisson pdf with parameter λ evaluated at x.

This distribution can be written as the expectation of $\mathbb{E}\varpi(z+X)$ where the variable X is distributed as a Poisson of parameter λ. An unbiased estimator is obtained by averaging over M_X samples of a Poisson distribution.

We can apply the methodology described above to this case (see Figure 1.2). We use a Gaussian proposal with moments calibrated from the method of moment estimator. In Figure 1.2 we show the effect of changing the value of M_X and hence modifying the variance of our unbiased estimator.

On a side note, notice that the estimator of the normalizing constant introduced above is unbiased. Hence it can be used inside other algorithm to perform inference on an intractable posterior, we use this approach for instance in Chapter 2, for model selection.

Resampling

Another important alternative is to sample from the empirical mixture formed by the weighted sample obtained with importance sampling,

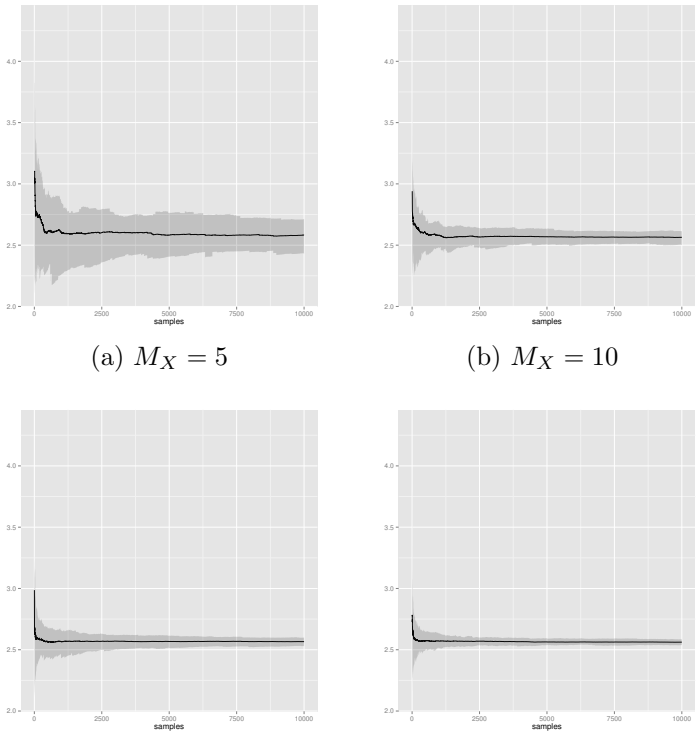
$$\hat{\mu}^{RIS}(\mathrm{d}x) = \sum_{i=1}^{M} \frac{w_i}{\sum_j w_j} \delta_{x_i}(\mathrm{d}x),$$

where $(w_i, x_i)_{1\leq\cdots\leq M}$ is a consistently weighted sample from the target distribution (i.e. the IS estimator converges to the correct distribution).

Several algorithms exist to perform resampling. In this thesis we will use systematic resampling (see Douc et al. [2005] for a comparison of resampling algorithms for particle filters). The convergence of the measure to the target in ensured by portmanteau lemma. Finite sample results also exist: Chapter 7 of Cappé et al. [2005] provides a Hoeffding type inequality.

Monte Carlo Markov Chain (MCMC)

MCMC relies on a different idea. It consists in building a Markov kernel with the target distribution as an invariant distribution. MCMC relies on the ergodic



(a) $M_X = 5$ (b) $M_X = 10$

(c) $M_X = 15$ (d) $M_X = 20$

Figure 1.2: Sequence of estimator of the parameter of a symmetric Skellam distribution

We provide the convergent sequence of estimator of the conditional expectation of the parameter for different value of M_X. We show the IS estimator and 95% confidence bands (obtained by replication). We use a simulated dataset for illustration with $n = 40$ observations. The value of M_X has as expected an influence on the variance.

---

theorem to approach integration with respect to the invariant distribution. That is, we rely on the following theorem.

Theorem 1.3.1 (Robert and Casella [2004b]) *Let $(X_n)_{n\geq 0}$ be a Harris recurrent Markov chain with invariant probability π for any f such that $f \in L_1(\pi)$ we have that*

$$\frac{1}{N}\sum_{i=1}^{N} f(X_i) \xrightarrow[N\to\infty]{} \int f(x)\pi(\mathrm{d}x).$$

As for the *iid* case we can get a CLT under additional assumptions. We refer the reader to Roberts and Rosenthal [2004] for a review on Markov chain theory

applied to MCMC and to Robert and Casella [2004b] for some applications of the results.

Gibbs sampler We start our very brief overview of MCMC methods with Gibbs sampling. The methodology consists in constructing a Markov Chain by sampling alternatively from conditional distributions. One can easily show that this results in a Markov Chain targeting the joint distribution.

We describe the algorithm in pseudo-code below

Algorithm 1 Two stage Gibbs sampler

Input: Conditional distributions $p(X|Z)$ and $p(Z|X)$, a starting point Z_0.

Output: $(X, Z)_{1 \leq t \leq T}$ a Markov chain with invariant probability $p(X, Z)$

For $t \in \{1, \ldots, T\}$

a. Sample $X_t \sim p(.|Z_{t-1})$.

b. Sample $Z_t \sim p(.|X_t)$.

End For

The Gibbs sampler needs conditional conjugacy in its simplest form to work. For our example 1.1.1 on the probit model we can see that putting a Gaussian prior and sampling alternatively from $p(\beta|Z_{1:n}, Y_{1:n})$ and $p(Z_{1:n}|\beta, Y_{1:n})$ leads to a sampler targeting the correct joint distribution $\pi(\beta, Z_{1:n}|X_{1:n})$. It is direct to see that the joint distribution of β, Z is given by,

$$\pi(\beta, Z_{1:n}|X_{1:n}) = \prod_{i=1}^{n} \left\{ \mathbb{1}_{Y_i=1} \mathbb{1}_{Z_i \geq 0} + \mathbb{1}_{Y_i=0} \mathbb{1}_{Z_i \leq 0} \right\} \varphi(Z_i; x_i\beta, 1) \varphi(\beta; 0_p, \vartheta I_p).$$

Hence the two conditionals are given by:

$$\begin{cases} \beta|(Z, Y)_{1:n} \sim \mathcal{N}\left(Q^{-1} x^t Y, Q^{-1}\right) \\ Z_i|\beta, Y \sim \mathcal{N}(x_i\beta, 1) \left\{ \mathbb{1}_{Y_i=1} \mathbb{1}_{Z_i \geq 0} + \mathbb{1}_{Y_i=0} \mathbb{1}_{Z_i < 0} \right\} \end{cases}$$

with $Q = \left(x^T x + \vartheta I_p\right)^{-1}$. One iterates between the two distributions as shown in Algorithm 1. In Chapter 2 we discuss alternatives to the Gibbs sampler that outperform it on many datasets.

Unbiased target In the case where only an unbiased version of the density is available, one can still construct a Gibbs sampler. Suppose that we can write the following joint distribution:

$$p(\theta, X_{1:M_X}) = \left(\frac{1}{M_X}\sum_{i=1}^{M_X}\hat{T}_{X_i}(\theta)\right)\prod_{i=1}^{M_X} q_i(X_i)$$

such that integrating with respect to X gives the correct target density (with q an auxiliary distribution). That is such that $\hat{T}(\theta)$ is an unbiased estimator under q of our target.

We can augment the space with an index k chosen with probability $\frac{\hat{T}_{X_k}(\theta)}{M_X}$. The joint distribution is then given by:

$$p(k, \theta, X_{1:M_X}) = \hat{T}_{X_k}(\theta)\frac{1}{M_X}\prod_{i=1}^{M_X} q(X_i)$$

Summing over k gives the correct distribution. Hence the Gibbs sampler constructed on this joint distribution, which samples iteratively from $k|\theta, X_{1:M_X}$, $\theta|k, X_k$ and $X_{1:M_X}|\theta, k$ gives the correct distribution.

This idea will be the building block of the parameter update in Chapter 6; the algorithm is developed in the special case of state space models. We use again the Skellam distribution as a toy example to illustrate the principle.

Example 1.3.3 *Symmetric Skellam distribution (continued)*

We put a $\Gamma(1,1)$ prior on λ as was done previously. Using what was described above we can write an algorithm to sample from the posterior. The efficiency of the algorithm could be discussed but it is however given here just as an example to prepare the reader for more advanced use of this idea in Chapter 6.

Algorithm 2 A Gibbs sampler for the Estimation of the Skellam parameter.

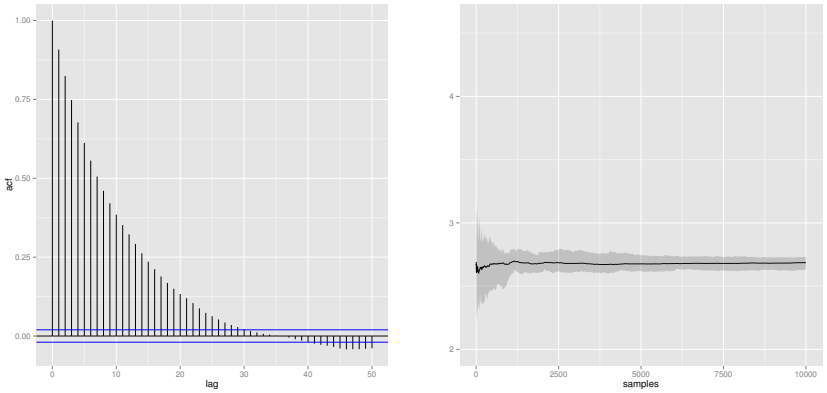
Input: λ_0, M_X and M_θ

Output: $(\lambda_i)_{0\leq i\leq M_\theta}$

For $i \in \{1, \cdots, M_\theta\}$

- **a.** Sample M_X i.i.d samples $X_{1:N}^j \sim \mathcal{P}^N(\lambda)$ for $j \neq k$
- **b.** Sample $k|\lambda, X_{1:n}^{1:M_X} \sim \sum_{i=1}^{M_X}\left(\prod_j \varpi(y_i + X_j^i; \lambda)\right)\delta_i$
- **c.** Sample $\lambda|X_{1:n}^k, y \sim \Gamma(2 + \sum_{i=1}^n (2y_i + X_i^k), 2n+1)$

End For



(a) ACF (b) $M_X = 130$

Figure 1.3: ACF and trace plot of the Gibbs sampler for the Skellam distribution

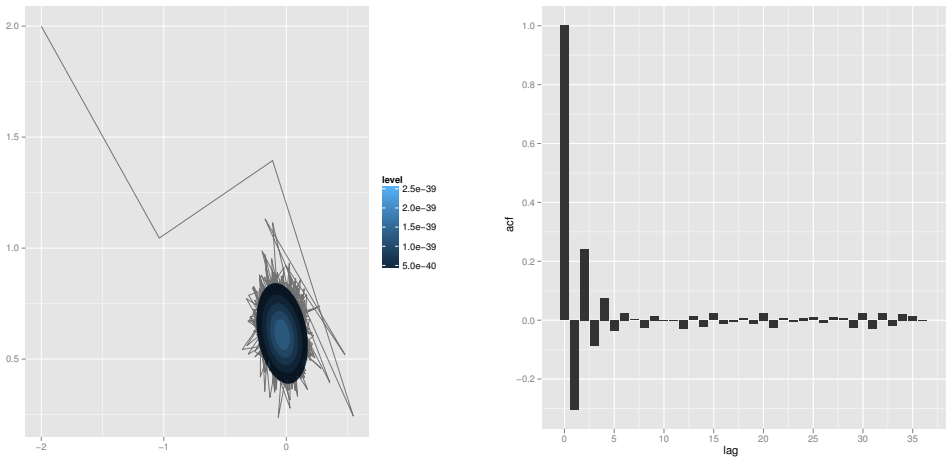
The experiment is performed on the same simulated dataset as for importance sampling. It can be compared to Figure 1.2. At first glance the scheme does not seem that efficient as we need a large M_X to get a variance comparable to the IS version.

Remark 1.3.2 *Before we start explaining a more general class of algorithms,let us note a property of the autocorrelations of Gibbs samplers.*

Assuming without loss of generality $\mathbb{E}h(X) = 0$*, we have that*

$$\begin{aligned}\mathbb{E}\left(h(X)h(X')\right) &= \int h(X)h(X') \int p(X|Z)p(Z|X')\mathrm{d}Zp(X')\mathrm{d}X'\mathrm{d}X \\ &= \int \left(\int h(X)p(X|Z)\mathrm{d}X\right)^2 p(Z)\mathrm{d}Z \geq 0\end{aligned}$$

This is not the case for Metropolis-Hastings. In fact, using a certain version of a MCMC algorithm, Hamiltonian Monte Carlo (HMC) (see Chapter 2) we can construct an example where the correlation is negative between samples therefore leading to a gain in efficiency as compared to iid sampling.



(a) (b)

Figure 1.4: An example of an MCMC run with negative autocorrelation

We show the trace plot for the two first marginals of the posterior of a probit model on the Pima dataset, the second panel shows the ACF of the first marginal.

Metropolis-Hasting A more general approach to building a Markov chain with the correct invariant distribution is the Metropolis-Hastings (MH) algorithm. The algorithm uses a proposal and acceptance mechanism. A new state is proposed according to a Markov kernel, with density q. The proposal is accepted with a given probability (see Algorithm 3). The degree of freedom of the algorithm is given by the choice of the proposal q. Several common choices have been adopted in the literature. We give a few. First one may choose to propose independently from the current state. The difficulty with the approach is that we need to ensure that the proposal dominates the target. By adopting this view we are back to the problem of AR and IS. A more useful approach is to move locally around the current point by using a random walk (i.e. a distribution such that $q(x|y) = q(y|x)$). The density of the proposal therefore cancels in the acceptance ratio.

Many other proposals will be considered in the sequel, in particular we will make use of the gradient of the target to explore more efficiently the state space.

Unbiased target As for IS one requirement of those methods is the availability of the density up to a multiplicative factor. Suppose we have access to a random variable $\hat{T}_X(\theta)$ such that $\mathbb{E}_q\left(\hat{T}_X(\theta)\right) = T(\theta)$ is our target distribution. By taking

Algorithm 3 Metropolis Hastings algorithm

Input: θ_0, M

Output: $(\theta_t)_{t\geq 0}$

For $t \in \{1, \cdots, M\}$

a. Sample $\theta_{\text{prop}} \sim q(.|\theta_{t-1})$.

b. Sample $U \sim \mathcal{U}([0,1])$.

c. If $U \leq \frac{\pi(\theta_{\text{prop}}|\mathcal{D})q(\theta_{t-1}|\theta_{\text{prop}})}{\pi(\theta_{t-1}|\mathcal{D})q(\theta_{\text{prop}}|\theta_{t-1})}$, set $\theta_t \leftarrow \theta_{\text{prop}}$, otherwise set $\theta_t \leftarrow \theta_{t-1}$.

End For

for proposal on (θ, X), $q(\theta'|\theta)q(X')$, and for target distribution $\hat{T}_X(\theta)q(X)$ one gets the correct invariant with the acceptance ratio:

$$\alpha(\theta, \theta') = 1 \wedge \frac{\hat{T}_{X'}(\theta')\cancel{q(X')}q(\theta|\theta')\cancel{q(X)}}{\hat{T}_X(\theta)\cancel{q(X)}q(\theta'|\theta)\cancel{q(X')}}.$$

In addition after simplification the acceptance ratio writes:

$$\alpha(\theta, \theta') = 1 \wedge \frac{\hat{T}_{X'}(\theta')q(\theta|\theta')}{\hat{T}_X(\theta)q(\theta'|\theta)}.$$

Hence applying the methodology of Algorithm 3 to an unbiased estimator of the target yields a Markov chain with the correct invariant probability.

Sequential Monte Carlo (SMC)

The first step to building the SMC algorithm is to define a sequence of densities $(\pi_n)_{\in\mathbb{T}}$ where $\mathbb{T}$ is an index set, and such that there is one $N \in \mathbb{T}$ such that π_N is the target. The idea is to apply IS to move from one distribution to another. We will explain in the rest of the section how we can take profit of resampling and ideas from MCMC to improve upon this. SMC is in particular useful for moderate dimensions in the case where MCMC explores the space too slowly.

We give a brief overview of the idea of the algorithm and relay the full description to Chapter 2 for static models and to Chapter 3 and 6 for SSM.

Particle filter (PF) Particle filters originated in Gordon et al. [1993] for state estimation in SSM. We describe the algorithm using the notations of Example 1.1.2.

The index set in this case is time and we want to sample recursively from the predictive distribution $p(x_{t+1}|y_{1:t})$ and the filtering distribution $p(x_t|y_{1:t})$. To this aim we introduce the forward backward recursion,

$$p(x_{t+1}|y_{1:t}) = \int_{\mathcal{X}} p(x_t|y_{1:t}) g(x_t|x_{t-1}) \mathrm{d}x_t$$
$$p(x_{t+1}|y_{1:t+1}) \propto p(x_{t+1}|y_{1:t}) f(y_{t+1}|x_{t+1})$$

This suggests the following importance sampling based algorithm: start by sampling an array of size M independently according to the initial distribution $x_i \sim g_0(x_0^i)$; we call the elements particles. Move the particles according to the Markov kernel $g(x_1^i|x_0^i)$ weight each particles according to $w_1^i = f_0(y_1|x_1)$. The operation is iterated this way for every time steps t. In the end we sample recursively from the Markov chain $\prod_{t=1}^T g(x_t|x_{t-1})$ and compute the product of weights $w_t^i = \prod_{t=1}^T f(y_t|x_t^i)$. This algorithm amounts to an importance sampling algorithm where the simulations and weight computation are done sequentially. It is referred in the literature as sequential importance sampling (SIS) [Cappé et al., 2005].

We can use the algorithm to get an unbiased estimator of the normalizing constant, for the case of SSM given a value of the parameters θ it corresponds to the likelihood, $\mathcal{L}(y_{1:T};\theta) = \int \prod_t p(y_t|x_t;\theta) g_t(x_t|x_{t-1};\theta) \mu_0(\mathrm{d}x_0) \mathrm{d}x_{0:T}$. In chapter 3 we reinterpret the GHK algorithm as a SIS algorithm for a given state space model, we use it to build a more efficient estimator of the Gaussian orthant probability.

The SIS algorithm as described proposes the state of the Markov chain using the *true* process $g_t(x_t|x_{t-1})$ at each time t. On can easily see, that as for IS, we can choose an auxiliary distribution as long as we correct accordingly in the weights. We will use this feature in Chapter 3 and in fact propose each state by the optimal proposal (minimizing the variance).

One issue of SIS is the fact that we have to compute a product of weights; this leads to weight degeneracy [Cappé et al., 2005]. As T grows the number of particles with small weights will grow, and the normalized weight will go to zero. One can measure the variance of the weigths by introducing the ESS (effective sample size) Kong et al. [1994],

$$ESS_t = \frac{\left\{\sum_{i=1}^M w_t^i\right\}^2}{\sum_{i=1}^M (w_t^i)^2} \in [0, M].$$

As the time horizon grows it is common to observe that this quantity goes to zero. In Chapter 3 we show that for a specific model, when computing the orthant probability with GHK, some quantity closely related to the ESS goes to 0 exponentially fast.

The solution proposed in the literature is to introduce resampling at each step or in an adaptive manner whenever the ESS falls under a given value. The particles are resampled according to their current weights and the later are set to one. We do not describe the case of PF with unbiased estimator of the weights although the interested reader can refer to Fearnhead et al. [2010].

Algorithm 4 Particle Filter

Input: M the number of particles
Sample: Sample $x_0^i \sim g_0(.)$
for $t = 1 : T-1$ **do**
  if $ESS < \eta^\star$ **then**
    $Z \leftarrow Z \times \{\frac{1}{M}\sum_{i=1}^M w_t^i\}$
    Resample $a_t^j \sim \sum_i \frac{w_t^i}{\sum_j w_t^j}\delta_i$, set $w_t^j \leftarrow 1$
  else
    $a_t^{1:M} = 1 : M$
  end if
  Sample $x_{t+1}^i \sim q_{t+1}(.|x_t^{a_t^i})$
  Set $w_{t+1}^i \leftarrow w_t^i \frac{f_{t+1}(y_{t+1}|x_{t+1}^i) g_{t+1}(x_{t+1}^i|x_t^{a_t^i})}{q_{t+1}(x_{t+1}^i|x_t^{a_t^i})}$
end for
return $Z \times \frac{1}{M}\sum_{i=1}^M w_T^i$ and $(x_T^i, w_T^i)_{1\leq i\leq M}$

Particle filters can also be used to compute the likelihood of the model. In fact it can be shown that the quantity defined by Z in Algorithm 4 is an unbiased estimator of the likelihood (Del Moral [1996a], Lemma 3). This is useful in particular in lights of the remarks made on the use of algorithms with unbiased estimator of the target. Andrieu et al. [2010a] suggested using a Gibbs sampler and a MH using the estimator of the likelihood given by a particle filter.

In the following we will show that this algorithm can be generalized to more general problems.

Static models The idea of using this framework to sample from other types of distribution originated in Neal [2001a], Chopin [2002a], (see also Del Moral et al. [2006b]).

The user needs to define a sequence of distribution; several choices can be considered. We describe the two used in this thesis.

- The first one was proposed under the name IBIS (iterated batch importance sampling) [Chopin, 2002a].

$$\pi_n(\theta) \propto \pi(\theta)\mathcal{L}(y_{1:n}|\theta)$$

and consists in adding one or several data points in a sequential manner. It is used in particular in Chapter 6 for parameter estimation of state space models.

- The sequence builds a bridge between two distributions,

$$\pi_n(\theta) \propto \{\pi(\theta)\mathcal{L}(y_{1:N}|\theta)\}^{\gamma_n} \{q(\theta)\}^{1-\gamma_n}$$

whith γ_n a sequence such that $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_N = 1$. The choice of the sequence is of paramount importance for the efficiency of the algorithm. In what follows we use an adaptive choice for γ such that the ESS does not fall under a given threshold (see Jasra et al. [2011b] and the different chapters of this thesis).

After a few resampling steps all particles θ would tend to be equal. We introduce an additional step to increase the diversity of the particle system, at time n we move the particles according to a kernel K_n that leaves π_n invariant. The choice of K_n is of paramount importance as it will condition the diversity of the particle system.

The main steps are described in Algorithm (12) bellow:

Algorithm 5 Sequential Monte Carlo

Input Γ,a, M, $\alpha^\star$, Set $Z \leftarrow 1$
Each computation involving m is done $\forall m \in 1:M$
Init $\eta_1^m \sim \pi_0(.)$, and $w_1^m = 1$
for $t \in 1:T-1$ **do**
  At time t the weighted system is distributed as $(w_t^m, \eta_{1:t}^m) \sim \pi_t(.)$.
  if $ESS(w_t^{1:M}) < \alpha^\star$ **then**
    $Z \leftarrow Z \times \{\frac{1}{M}\sum_{i=1}^M w_t^i\}$.
    Resample: $\eta_t'^m \sim \sum_{j=1}^M w_t^j \delta_{\eta_t^j}$, $w_t^m \leftarrow 1$.
    Move: $\eta_t^m \sim K_t(\eta_t'^m, d\eta_t^m)$ where K_t leaves $\pi_t(\eta_{1:t})$ invariant.
  end if
  $w_{t+1}^m \leftarrow w_t^m \times \frac{\pi_{t+1}(\eta_t^m)}{\pi_t(\eta_t^m)}$.
end for
return $Z \times \{\frac{1}{M}\sum_{i=1}^M w_T^i\}$, and (w_T^i, η_T^i)

A nice feature of SMC is that at a given step one can use estimators based on the system of particles to build adaptive proposals and MCMC steps. We will make some use of this in the different chapters, see in particular Chapter 6 where we use SMC in the context of intractable targets.

1.3.2 Approximate Inference

In the previous section we justified the algorithms as being exact with the CPU effort growing to infinity. Another approach is to approximate the target by the closest distribution in a tractable family. We give a brief overview of the two most used algorithms of this type and relay a full description to Chapter 2 for expectation propagation (EP) and Chapter 4 for variational Bayes (VB).

Expectation Propagation

In most applications of EP we will consider a posterior that can be written as a product of a tractable distribution and intractable factors. We call tractable a distribution for which we have easy access to moments and is stable by multiplication (as are exponential family models). We write:

$$\pi(\theta|Y) \propto \pi(\theta) \prod_i t_i(\theta).$$

EP works by successively approximating each sites by a given parametric distribution. To be more specific let us describe the global approximation Q to the previous posterior as a product:

$$Q(\theta) \propto \pi(\theta) \prod_i \tilde{t}_i(\theta),$$

where each $\tilde{t}_i(\theta)$ approximates its corresponding site $t_i(\theta)$. EP works by cycling through the sites and updating them. Once a site has been updated we can use this to update the global approximation. Let us describe the process in more detail. We start by computing the cavity distribution as the approximation removed of the influence of site j,

$$Q^{\setminus j}(\theta) \propto \pi(\theta) \prod_{i \neq j} \tilde{t}_i(\theta).$$

This distribution is easily obtained for exponential models, and consists only in a modification of the natural parameters. We multiply the cavity distribution by the corresponding new site $t_j(\theta)$ to obtain the hybrid distribution:

$$h_j(\theta) = \frac{Q^{\setminus j}(\theta) t_j(\theta)}{\int Q^{\setminus j}(\theta) t_j(\theta) \mathrm{d}\theta}.$$

This new distribution would ideally be our new distribution updating information of site j, it is however intractable. The approach consists in defining the new approximation as the one minimizing the KL divergence of Q with respect to the hybrid over the considered exponential family. Because the approximating

distribution is in an exponential family the minimization can be done by computing the moments of the distribution (see Seeger [2005a]). This is done in cyclic manner for each site until a convergence criterion is achieved. We give an example where an EP algorithm can be used, and an iteration of the algorithm for a simple case.

Example 1.3.4 *PAC-Bayesian 0-1 Classification The pseudo posterior we want to approximate is given by:*

$$\pi_{\xi,\gamma}(\theta|\mathcal{D}) \propto \prod_{i=1}^{n} \exp\left(-\frac{\lambda}{n}\mathbb{1}_{Y_i<X_i,\theta>\leq 0}\right)\pi(\theta)$$

and we suppose that the prior $\pi(\theta)$ is Gaussian.

We can use EP to get a Gaussian approximation of the posterior because the moments of the distribution $h_i(\theta) \propto \varphi(\theta; m^{\setminus i}, \Sigma^{\setminus i}) \exp\left(-\frac{\lambda}{n}\mathbb{1}_{Y_i<X_i,\theta>\leq 0}\right)$ can be computed exactly.

To make this application clearer, in Figure 1.5 we show the iterations of an EP approximation for a site of the form $t(\theta) = \mathbb{1}_{\theta<0}$.

More details on the computation of each terms in specific cases are given in the chapters of this thesis. Note in particular the EP is used in Chapter 2 to compute an approximation of the posterior of a probit model, and in Chapter 4 to approximate the Gibbs posterior under a AUC loss.

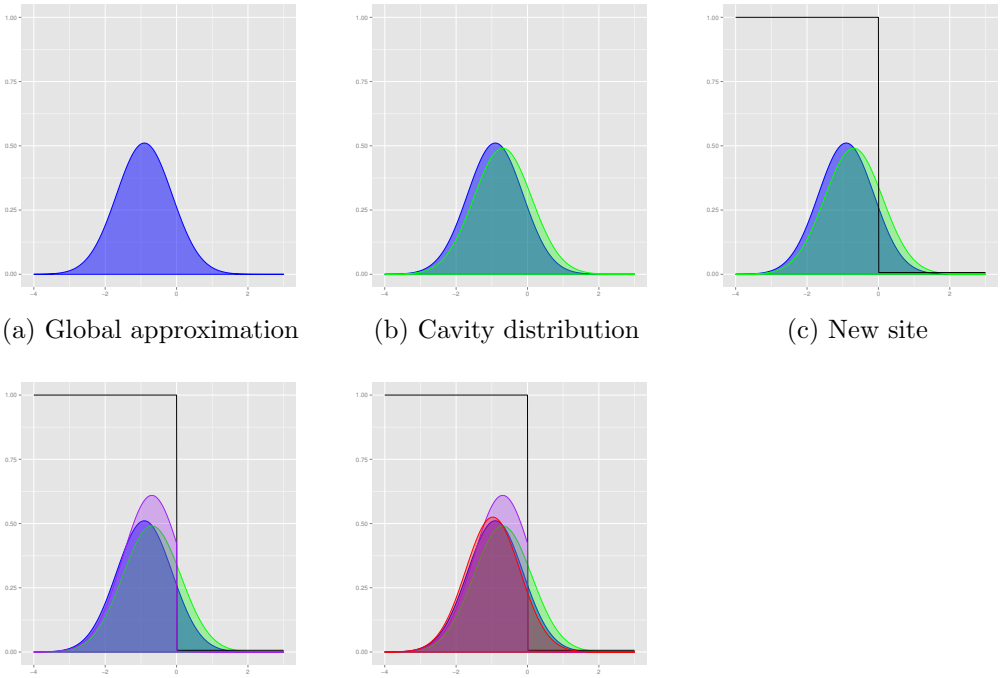
Fractional Expectation Propagation

Fractional EP is similar to the algorithm described previously with the exception that only a fraction of each site is treated at each iteration. We will denote this fraction by $\alpha \leq 1$. The corresponding cavity distribution is now given by $Q^{\setminus\alpha,j}(\theta) \propto \prod_{i=1}^{n} \tilde{t}_i(\theta)\left(\tilde{t}_j(\theta)\right)^{1-\alpha}$. The hybrid distribution can be written has $h_j(\theta) \propto Q^{\setminus\alpha,j}(\theta)\left(t_j(\theta)\right)^{\alpha}$. The rest of the algorithm follows the steps of standard EP.

The rationale behind this strategy is twofold [Jylänki et al., 2011]. First taking $\alpha < 1$ for non log-concave sites flattens the distribution and therefore prevents numerical issues linked to multimodality. Second it prevents the cavity moment of becoming too small (or even negative). Taking specific values of α can also allow us to transform a intractable problem into a tractable one. We give such an example in the following.

Example 1.3.5 *Consider the Gaussian regression model with Student prior distribution as advocated for instance by Gelman et al. [2008].*

$$\pi(\beta|Y) \propto \exp\left(-\frac{1}{2}\|Y - X^t\beta\|_2^2\right)\prod_{i=1}^{d} t_\nu(\beta_i)$$



(a) Global approximation (b) Cavity distribution (c) New site

(d) Hybrid distribution (e) New global approximation

Figure 1.5: Expectation Propagation in Action

In panel (a) we show the global approximation Q (initial point), the site approximation is removed to get $Q^{\setminus j}$ in panel (b). Panel (c) and (d) show the site (indicator function) and the hybrid $h(\theta) \propto \mathcal{N}(\theta; m, s)\mathbb{1}_{\theta>0}$ (in purple). The closest Gaussian to the hybrid is the new approximation panel (e) (the one with the same mean and variance).

---

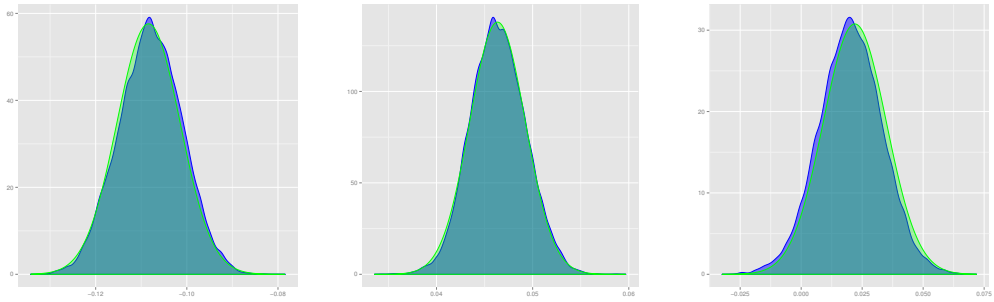
where $t_{\nu,\gamma}(\beta_i)$ is the Student density with degree of freedom ν and scale γ. This prior leads to an intractable problem.

Standard EP algorithm also leads to intractable integrals using fractional EP we can get a tractable problem that leads to tractable sites.

We can write

$$t_{\nu,\gamma}(\beta_i) \propto \frac{1}{\left(1 + \left(\frac{\beta_i}{\gamma}\right)^2\right)^{\frac{\nu+1}{2}}}$$

Taking a fractional parameter $\eta = -\frac{2}{\nu+1}$ for a Gaussian approximation one gets a tractable EP approximation for an otherwise intractable case. This can also be used in the Gaussian process case with Student likelihood to prevent from using numerical integration [Jylänki et al., 2011].



(a) β_1 (b) β_2 (c) β_3

Figure 1.6: Marginal of the Gaussian approximation of a regression with Student priors

Regression with student prior marginal of the 3 first coefficients on the Boston dataset. In green the EP approximation and in blue the marginal from the sampled posterior (RWMH).

Variational Bayes

The variational Bayes algorithm appears as a way to transform an integral form problem in a finite dimension optimization problem (Jordan et al. [1999],MacKay [2002] and Chap. 10 in Bishop [2006a]).

The idea of the approach is to minimize the KL divergence of a measure ρ with respect to the posterior measure of interest. In most cases of interest it is an intractable problem because of the unknown normalizing constant, however we can write the following useful decomposition of the log marginal likelihood,

$$\log m(\mathcal{D}) = \mathbb{E}_\rho \left(\log \frac{\pi(\theta)\mathcal{L}(\mathcal{D}|\theta)}{\rho(\theta)} \right) + \mathcal{K}(\rho|\pi(.|\mathcal{D})).$$

Because the left handside does not depend on ρ, minimizing the first term of the right handside is equivalent to minimizing the KL divergence. This surrogate objective function can be seen as a lower bound of the marginal likelihood. The above definition gives a natural optimization algorithm, minimizing the objective on all probability distribution yields the posterior, we minimize the objective restricted to a set of tractable distributions.

We now review two types of families popular in the VB literature.

- Mean field VB: for a certain decomposition $\Theta = \Theta_1 \times \cdots \times \Theta_d$, $\mathcal{F}$ is the set

of product probability measures

$$\mathcal{F}^{\mathrm{MF}} = \left\{ \rho \in \mathcal{M}_+^1(\Theta) : \rho(\mathrm{d}\theta) = \prod_{i=1}^{d} \rho_i(\mathrm{d}\theta_i), \forall i \in \{1, \ldots, d\}, \rho_i \in \mathcal{M}_+^1(\Theta_i) \right\}. \tag{1.4}$$

The infimum of the KL divergence $\mathcal{K}(\rho, \pi(.|\mathcal{D}))$, relative to $\rho = \prod_i \rho_i$ satisfies the following fixed point condition [Bishop, 2006a; Parisi, 1988, Chap. 10]:

$$\forall j \in \{1, \cdots, d\} \quad \rho_j(\mathrm{d}\theta_j) \propto \exp\left(\int \{\log \mathcal{L}(\mathcal{D}|\theta) + \log \pi(\theta)\} \prod_{i \neq j} \rho_i(\mathrm{d}\theta_i) \right). \tag{1.5}$$

This leads to a natural algorithm were we update successively every ρ_j until stabilization.

- Parametric family:

$$\mathcal{F}^{\mathrm{P}} = \left\{ \rho \in \mathcal{M}_+^1(\Theta) : \rho(\mathrm{d}\theta) = f(\theta; m)\mathrm{d}\theta, m \in M \right\};$$

and M is finite-dimensional; say $\mathcal{F}^{\mathrm{P}}$ is the family of Gaussian distributions (of dimension d). In this case, several methods may be used to compute the infimum. As above, one may use fixed-point iterations, provided an equation similar to (5.4) is available. Alternatively, one may directly maximize the bound with respect to parameter m, using numerical optimization routines.

Mean field VB has been shown to underestimate the variance of the distribution. This is partly due to the fact the KL divergence takes infinite values if the target does not dominate ρ. In particular the condition is violated if the tails of ρ are lighter than the target.

To illustrate this problematic behavior let us look at an experiment proposed in Bishop [2006a], namely the mean field approximation of a centered bivariate Gaussian with variance 1 and correlation ρ. One readily checks that the iterations are given by variance of $1 - \rho^2$ and a sequence of means converging to 0. Figure 1.7 illustrates the two distributions, with the approximation constrained by the variance in the direction of the smallest eigenvalue.

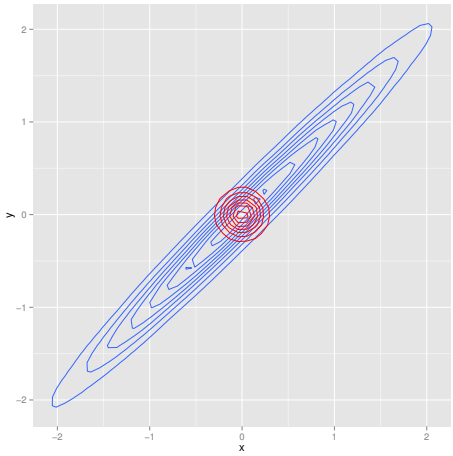


Figure 1.7: Mean field VB approximation (red) of bivariate Gaussian distribution with correlation $\rho = 0.99$ (blue)

In chapter 5 we give conditions under which VB approximations of the Gibbs posterior do not deteriorate the rate of convergence of the approximation.

1.4 Overview

This thesis is divided into 5 chapters, we give a short descriprion of each of them in the following.

1.4.1 Leave Pima indians alone: binary regression as a benchmark for Bayesian computation

Resumé Quand un nouvel algorithme est introduit en statistique bayésienne le modèle probit sur de petits jeux de données est fréquemment employé pour le tester. Ce chapitre étudie le bien fondé de cette approche. Il donne, de plus, une revue de la littérature dans le domaine avec pour exemple ce modèle. Les algorithmes étudiés sont divisés en deux catégories: d'un côté ceux employant des méthodes d'échantillonnage (échantillonnage préferentiel, méthode de Monte Carlo par chaîne de Markov et methode de Monte Carlo séquentielle), de l'autre les algorithmes d'approximation rapide (Laplace et EP). De nombreux résultats numériques sont présentés.

Abstract Whenever a new approach to perform Bayesian computation is introduced, a common practice is to showcase this approach on a binary regression model and datasets of moderate size. This chapter discusses to which extent this practice is sound. It also reviews the current state of the art of Bayesian computation, using binary regression as a running example. Both sampling-based algorithms (importance sampling, MCMC and SMC) and fast approximations (Laplace and EP) are covered. Extensive numerical results are provided, some of which might go against conventional wisdom regarding the effectiveness of certain algorithms. Implications for other problems (variable selection) and other models are also discussed.

1.4.2 Computation of Gaussian orthant probabilities in high dimension

Resumé Nous étudions dans ce chapitre le calcul de probabilités d'orthants (la probabilité qu'une réalisation Gaussienne ait toutes ses composantes positives). Nous basons cette étude sur l'algorithme GHK couramment utilisé pour résoudre ce problème en dimensions plus grandes que 10. Dans ce chapitre nous montrons, pour des matrices de variance-covariance markoviennes que l'algorithme peut s'interpréter comme un algorithme d'échantillonnage préférentiel séquentiel (SIS pour acronyme anglais). Pour un AR(1) la variance normalisée de GHK diverge à une vitesse exponentielle avec la dimension. Pour corriger ce problème nous introduisons une étape de réechantillonnage transformant ainsi l'algorithme en un filtre particulaire. Nous généralisons dans un deuxième temps cette idée au cas de matrices de variance covariance génerales en introduisant un algorithme de Monte Carlo séquentiel.

Abstract We study the computation of Gaussian orthant probabilities, i.e. the probability that a Gaussian variable falls inside a quadrant. The Geweke-Hajivassiliou-Keane (GHK) algorithm [Genz, 1992; Geweke, 1991; Hajivassiliou et al., 1996; Keane, 1993], is currently used for integrals of dimension greater than 10. In this chapter we show that for Markovian covariances GHK can be interpreted as the estimator of the normalizing constant of a state space model using sequential importance sampling (SIS). We show for an AR(1) the variance of the GHK, properly normalized, diverges exponentially fast with the dimension. As an improvement we propose using a particle filter (PF). We then generalize this idea to arbitrary covariance matrices using Sequential Monte Carlo (SMC) with properly tailored MCMC moves. We show empirically that this can lead to drastic improvements on currently used algorithms. We also extend the framework to orthants of mixture of Gaussians (Student, Cauchy etc.), and to the simulation of truncated Gaussians.

1.4.3 Theoretical and computational aspects of PAC Bayesian ranking and scoring

Resumé Nous développons une procédure d'ordonancement et de classification basée sur une approche PAC-Bayesienne et sur la minimisation d'un critère AUC. Nous démontrons des inegalités oracles pour différentes distributions *a priori* : un *prior* Gaussien et un *prior* "spike and slab". Dans un deuxième temps nous proposons deux algorithmes pour calculer l'estimateur numériquement. D'abord un algorithme de Monte Carlo sequentiel comme méthode "exacte" au sens de Monte Carlo, puis un algorithme d'approximation EP pour introduire des méthodes plus rapides.

Abstract We develop a scoring and classification procedure based on the PAC-Bayesian approach and the AUC (Area Under Curve) criterion. We focus initially on the class of linear score functions. We derive PAC-Bayesian non-asymptotic bounds for two types of prior for the score parameters: a Gaussian prior, and a spike and slab prior; the latter makes it possible to perform feature selection. One important advantage of our approach is that it is amenable to powerful Bayesian computational tools. We derive in particular a Sequential Monte Carlo algorithm, as an efficient method which may be used as a gold standard, and an expectation propagation algorithm, as a much faster but approximate method. We also extend our method to a class of non-linear score functions, essentially leading to a nonparametric procedure, by considering a Gaussian process prior.

1.4.4 Properties of variational approximations of Gibbs posteriors

Resumé L'approche PAC-bayésienne permet le développement de bornes non assymptotiques sur le risque théorique. La distribution sur les estimateurs qui en découlent n'est cependant pas calculable. Alors que l'approche usuelle est d'utiliser des méthodes de Monte Carlo par chaîne de Markov nous proposons pour plus d'efficacité une approximation variationelle des estimateurs. Nous étudions alors les propriétés de ces estimateurs. Nous donnons des conditions sous lesquelles l'approximation variationelle converge à la même vitesse que les procédures PAC usuellement considérées. Nous appliquons ensuite ces résultats à differents problèmes d'apprentissage (classification, ranking et completion de matrices). L'implémentation des algorithmes est également abordée.

Abstract The PAC-Bayesian approach is a powerful set of techniques to derive non-asymptotic risk bounds for random estimators. The corresponding optimal

distribution of estimators, usually called the Gibbs posterior, is unfortunately intractable. One may sample from it using Markov chain Monte Carlo, but this is often too slow for big datasets. We consider instead variational approximations of the Gibbs posterior, which are fast to compute. We undertake a general study of the properties of such approximations. Our main finding is that such a variational approximation has often the same rate of convergence as the original PAC-Bayesian procedure it approximates. We specialise our results to several learning tasks (classification, ranking, matrix completion), discuss how to implement a variational approximation in each case, and illustrate the good properties of said approximation on real datasets.

1.4.5 Towards automatic calibration of SMC^2

Resumé Nous étudions SMC^2, un algorithme pour l'estimation de paramètres dans les modèles à espace d'état. Nous génerons pour cela N_θ particules θ_m et pour chacune d'elle nous lançons un filtre particulaire de taille N_x (i.e. pour chaque pas de temps, N_x particules sont générées dans l'espace d'état $\mathcal{X}$). Le but de ce chapitre est d'automatiser le choix de N_x au cours de l'algorithme. Nous utilisons pour cela un algorithme de Monte Carlo séquentiel conditioné à une trajectoire. Pour réduire le coût mémoriel nous proposons de sauvegarder l'état initial des générateurs pseudo-aléatoires pour chaque filtre particulaire. Le choix de N_x est conditioné à un estimateur de la variance de l'estimateur sans biais de la vraisemblance obtenu par des techniques non parametriques. Les applications numériques sont effectuées à coût computationel constant et débouchent sur une plus petite erreur de Monte Carlo que l'algorithme original.

Abstract SMC^2 (Chopin et al., 2013) is an efficient algorithm for sequential estimation and state inference of state-space models. It generates N_θ parameter particles θ_m , and, for each θ^m , it runs a particle filter of size N_x (i.e. at each time step, N_x particles are generated in the state space $\mathcal{X}$). We discuss how to automatically calibrate N_x in the course of the algorithm. Our approach relies on conditional Sequential Monte Carlo updates, monitoring the state of the pseudo random number generator and on an estimator of the variance of the unbiased estimate of the likelihood that is produced by the particle filters, which is obtained using nonparametric regression techniques. We observe that our approach is both less CPU intensive and with smaller Monte Carlo errors than the initial version of SMC^2 .

2

Leave Pima indians alone: binary regression as a benchmark for Bayesian computation

This is joint work with Nicolas Chopin
Status: submitted to *Statistical Science*

2.1 Introduction

The field of Bayesian computation seems hard to track these days, as it is blossoming in many directions. MCMC (Markov chain Monte Carlo) remains the main approach, but it is no longer restricted to Gibbs sampling and Hastings-Metropolis, as it includes more advanced, Physics-inspired methods, such as HMC [Hybrid Monte Carlo, Neal, 2010b] and its variants [Girolami and Calderhead, 2011; Hoffman and Gelman, 2013; Shahbaba et al., 2011]. On the other hand, there is also a growing interest for alternatives to MCMC, such as SMC (Sequential Monte Carlo, e.g. Del Moral et al., 2006a), nested sampling [Skilling, 2006], or the fast approximations that originated from machine learning, such as Variational Bayes [e.g. Bishop, 2006b, Chap. 10], and EP [Expectation Propagation, Minka, 2001b]. Even Laplace approximation has resurfaced in particular thanks to the INLA methodology [Rue et al., 2009].

One thing however that all these approaches have in common is they are almost always illustrated by a binary regression example; see e.g. the aforementioned papers. In other words, binary regressions models, such as probit or logit, are a de facto benchmark for Bayesian computation.

This remark leads to several questions. Are binary regression models a reasonable benchmark for Bayesian computation? Should they be used then to develop a 'benchmark culture' in Bayesian computation, like in e.g. optimisation? And

practically, which of these methods actually 'works best' for approximating the posterior distribution of a binary regression model?

The objective of this chapter is to answer these questions. As the ironic title suggests, our findings shall lead to us be critical of certain current practices. Specifically, most papers seem content with comparing some new algorithm with Gibbs sampling, on a few small datasets, such as the well-known Pima Indians diabetes dataset (8 covariates). But we shall see that, for such datasets, approaches that are even more basic than Gibbs sampling are actually hard to beat. In other words, datasets considered in the literature may be too toy-like to be used as a relevant benchmark. On the other hand, if ones considers larger datasets (with say 100 covariates), then not so many approaches seem to remain competitive.

We would also like to discuss *how* Bayesian computation algorithms should be compared. One obvious criterion is the error versus CPU time trade-off; this implies discussing which posterior quantities one may need to approximate. A related point is whether the considered method comes with a simple way to evaluate the numerical error. Other criteria of interest are: (a) how easy to implement is the considered method? (b) how generic is it? (does changing the prior or the link function requires a complete rewrite of the source code?) (c) to which extent does it require manual tuning to obtain good performance? (d) is it amenable to parallelisation? Points (a) and (b) are rarely discussed in Statistics, but relate to the important fact that, the simpler the program, the easier it is to maintain, and to make it bug-free. Regarding point (c), we warn beforehand that, as a matter of principle, we shall refuse to manually tune an algorithm on a per dataset basis. Rather, we will discuss, for each approach, some (hopefully reasonable) general recipe for how to choose the tuning parameters. This has two motivations. First, human time is far more valuable than computer time: Cook [2014] mentions that one hour of CPU time is today three orders of magnitude less expensive than one hour of pay for a programmer (or similarly a scientist). Second, any method requiring too much manual tuning through trial and error may be practically of no use beyond a small number of experts.

Finally, we also hope this chapter may serve as an up to date review of the state of Bayesian computation. We believe this review to be timely for a number of reasons. First, as already mentioned, because Bayesian computation seems to develop currently in several different directions. Second, and this relates to criterion (d), the current interest in parallel computation [Lee et al., 2010; Suchard et al., 2010] may require a re-assessment of Bayesian computational methods: method A may perform better than method B on a single core architecture, while performing much worse on a parallel architecture. Finally, although the phrase 'big data' seems to be a tired trope already, it is certainly true that datasets are getting bigger and bigger, which in return means that statistical methods needs to

be evaluated on bigger and bigger datasets. To be fair, we will not really consider in this work the kind of huge datasets that pertain to 'big data', but we will at least strive to move away from the kind of 'ridiculously small' data encountered too often in Bayesian computation papers.

The chapter is structured as follows. Section 2.2 covers certain useful preliminaries on binary regression models. Section 2.3 discusses fast approximations, that is, deterministic algorithms that offer an approximation of the posterior, at a lower cost than sampling-based methods. Section 2.4 discusses 'exact', sampling-based methods. Section 2.5 is the most important part of the chapter, as it contains an extensive numerical comparison of all these methods. Section 2.6 discusses variable selection. Section 2.7 discusses our findings, and their implications for both end users and Bayesian computation experts.

2.2 Preliminaries: binary regression models

2.2.1 Likelihood, prior

The likelihood of a binary regression model have the generic expression

$$p(\mathcal{D}|\boldsymbol{\beta}) = \prod_{i=1}^{n_{\mathcal{D}}} F(y_i \boldsymbol{\beta}^T \boldsymbol{x}_i) \tag{2.1}$$

where the data $\mathcal{D}$ consist of n responses $y_i \in \{-1, 1\}$ and n vectors $\boldsymbol{x}_i$ of p covariates, and F is some CDF (cumulative distribution function) that transforms the linear form $y_i \boldsymbol{\beta}^T \boldsymbol{x}_i$ into a probability. Taking $F = \Phi$, the standard normal CDF, gives the probit model, while taking $F = L$, the logistic CDF, $L(x) = 1/\left(1 + e^{-x}\right)$, leads to the logistic model. Other choices could be considered, such as e.g. the CDF of a Student distribution (robit model) to better accommodate outliers.

We follow Gelman et al. [2008]'s recommendation to standardise the predictors in a preliminary step: non-binary predictors have mean 0 and standard deviation 0.5, binary predictors have mean 0 and range 1, and the intercept (if present) is set to 1. This standardisation facilitates prior specification: one then may set up a "weakly informative" prior for $\boldsymbol{\beta}$, that is a proper prior that assigns a low probability that the marginal effect of one predictor is outside a reasonable range. Specifically, we shall consider two priors $p(\boldsymbol{\beta})$ in this work: (a) the default prior recommended by Gelman et al. [2008], a product of independent Cauchys with centre 0 and scale 10 for the constant predictor, 2.5 for all the other predictors (henceforth, the Cauchy prior); and (b) a product of independent Gaussians with mean 0 and standard deviation equal to twice the scale of the Cauchy prior (henceforth the Gaussian prior).

Of course, other priors could be considered, such as e.g. Jeffreys' prior [Firth, 1993], or a Laplace prior [Kabán, 2007]. Our main point in considering the two priors above is to determine to which extent certain Bayesian computation methods may be prior-dependent, either in their implementation (e.g. Gibbs sampling) or in their performance, or both. In particular, one may expect the Cauchy prior to be more difficult to deal with, given its heavy tails.

2.2.2 Posterior maximisation (Gaussian prior)

We explain in this section how to quickly compute the mode, and the Hessian at the mode, of the posterior:

$$p(\boldsymbol{\beta}|\mathcal{D}) = \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\mathcal{D})}, \quad p(\mathcal{D}) = \int_{\mathbb{R}^d} p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\, d\boldsymbol{\beta},$$

where $p(\boldsymbol{\beta})$ is one of the two priors presented in the previous section, and $Z(\mathcal{D})$ is the marginal likelihood of the data (also known as the evidence). These quantities will prove useful later, in particular to tune certain of the considered methods.

The two first derivatives of the log-posterior density may be computed as:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}|\mathcal{D}) &= \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}) + \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\mathcal{D}|\boldsymbol{\beta}), \\
\frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} \log p(\boldsymbol{\beta}|\mathcal{D}) &= \frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} \log p(\boldsymbol{\beta}) + \frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} \log p(\mathcal{D}|\boldsymbol{\beta})
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\mathcal{D}|\boldsymbol{\beta}) &= \sum_{i=1}^{n_\mathcal{D}} (\log F)' (y_i \boldsymbol{\beta}^T \boldsymbol{x}_i) y_i \boldsymbol{x}_i \\
\frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} \log p(\mathcal{D}|\boldsymbol{\beta}) &= \sum_{i=1}^{n_\mathcal{D}} (\log F)'' (y_i \boldsymbol{\beta}^T \boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{x}_i^T
\end{aligned}$$

and $(\log F)'$ and $(\log F)''$ are the two first derivatives of $\log F$. Provided that $\log F$ is concave, which is the case for probit and logit regressions, the Hessian of the log-likelihood is clearly a negative definite matrix. Moreover, if we consider the Gaussian prior, then the Hessian is of the log-posterior is also negative (as the sum of two negative matrices, as Gaussian densities are log-concave). We stick to the Gaussian prior for now.

This suggests the following standard approach to compute the MAP (maximum a posterior) estimator, that is the point $\boldsymbol{\beta}_{\text{MAP}}$ that maximises the posterior density $p(\boldsymbol{\beta}|\mathcal{D})$: to use Newton-Raphson, that is, to iterate

$$\boldsymbol{\beta}_{(\text{new})} = \boldsymbol{\beta}_{(\text{old})} - \boldsymbol{H}^{-1} \left\{ \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}_{(\text{old})}|\mathcal{D}) \right\} \tag{2.2}$$

until convergence is reached; here $\boldsymbol{H}$ is Hessian of the log posterior at $\boldsymbol{\beta} = \boldsymbol{\beta}_{(\text{old})}$, as computed above. The iteration above corresponds to finding the zero of a local, quadratic approximation of the log-posterior. Newton-Raphson typically works very well (converges in a small number of iterations) when the function to maximise is concave.

We note two points in passing. First, one may obtain the MLE (maximum likelihood estimator) by simply taking $p(\boldsymbol{\beta}) = 1$ above (i.e. a Gaussian with infinite variance). But the MLE is not properly defined when complete separation occurs, that is, there exists a hyperplane that separates perfectly the two outcomes: $y_i \boldsymbol{\beta}_{\text{CS}}^T \boldsymbol{x}_i \geq 0$ for some $\boldsymbol{\beta}_{\text{CS}}$ and all $i \in 1 : N$. This remark gives an extra incentive for performing Bayesian inference, or at least MAP estimation, in cases where complete separation may occur, in particular when the number of covariates is large [Firth, 1993; Gelman et al., 2008].

Second, $\boldsymbol{H}$ in (2.2) is sometimes replaced by some approximation, leading to so-called quasi-Newton algorithms. Some of these algorithms such as IRLS (iterated reweighted least squares) have a nice statistical interpretation. For our purposes however, all these variants seem to show similar performance, so we will stick to the standard version of Newton-Raphson.

2.2.3 Posterior maximisation (Cauchy prior)

The log-density of the Cauchy prior is not concave:

$$\log p(\boldsymbol{\beta}) = -\sum_{j=1}^{p} \log(\pi\sigma_j) - \sum_{j=1}^{p} \log(1 + \beta_j^2/\sigma_j^2)$$

for scales σ_j chosen as explained in Section 2.2.1. Hence, the corresponding log-posterior is no longer guaranteed to be concave, which in turn means that Newton-Raphson algorithm might fail to converge.

However, we shall observe that, for most of the datasets considered in this chapter, Newton-Raphson does converge quickly even for our Cauchy prior. In each case, we used as starting point for the Newton-Raphson iterations the OLS (ordinary least square) estimate. We suspect what happens is that, for most standard datasets, the posterior derived from a Cauchy prior remains log-concave, at least in a region that encloses the MAP estimator and our starting point.

2.3 Fast approximation methods

This section discusses fast approximation methods, that is methods that are deterministic, fast (compared to sampling-based methods), but which comes with an approximation error which is difficult to assess. These methods include the Laplace approximation, which was popular in Statistics before the advent of MCMC methods, but also recent Machine Learning methods, such as EP (Expectation Propagation, Minka, 2001b), and VB (Variational Bayes, e.g. Bishop, 2006b, Chap. 10). However, we will not discuss VB, as Consonni and Marin [2007] give convincing (formal and numerical) arguments that this type of approach does not work well for probit models.

Concretely, we will focus on the approximation of the following posterior quantities: the marginal likelihood $p(\mathcal{D})$, as this may be used in model choice; and the marginal distributions $p(\beta_i|\mathcal{D})$ for each component β_i of $\boldsymbol{\beta}$. Clearly these are the most commonly used summaries of the posterior distribution, and other quantities, such as the posterior expectation of $\boldsymbol{\beta}$, may be directly deduced from them.

Finally, one should bear in mind that such fast approximations may be used as a preliminary step to calibrate an exact, more expensive method, such as those described in Section 2.4.

2.3.1 Laplace approximation

The Laplace approximation is based on a Taylor expansion of the posterior log-density around the mode $\boldsymbol{\beta}_{\text{MAP}}$:

$$\log p(\boldsymbol{\beta}|\mathcal{D}) \approx \log p(\boldsymbol{\beta}_{\text{MAP}}|\mathcal{D}) - \frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}}\right)^T \boldsymbol{Q}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}}\right),$$

where $\boldsymbol{Q} = -\boldsymbol{H}$, i.e. minus the Hessian of $\log p(\boldsymbol{\beta}|\mathcal{D})$ at $\boldsymbol{\beta} = \boldsymbol{\beta}_{\text{MAP}}$; recall that we explained how to compute these quantities in Section 2.2.2. One may deduce a Gaussian approximation of the posterior by simply exponentiating the equation above, and normalising:

$$\begin{aligned} q_L(\boldsymbol{\beta}) &= N_p\left(\boldsymbol{\beta}; \boldsymbol{\beta}_{\text{MAP}}, \boldsymbol{Q}^{-1}\right) \\ &:= (2\pi)^{-p/2} \left|\boldsymbol{Q}\right|^{1/2} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}}\right)^T \boldsymbol{Q}\left(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}}\right)\right\}. \quad (2.3) \end{aligned}$$

In addition, since for any $\boldsymbol{\beta}$,

$$p(\mathcal{D}) = \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathcal{D})}$$

one obtains an approximation to the marginal likelihood $p(\mathcal{D})$ as follows:

$$p(\mathcal{D}) \approx Z_L(\mathcal{D}) := \frac{p(\boldsymbol{\beta}_{\mathrm{MAP}})p(\mathcal{D}|\boldsymbol{\beta}_{\mathrm{MAP}})}{(2\pi)^{-p/2}\left|\boldsymbol{Q}\right|^{1/2}}.$$

From now on, we will refer to this particular Gaussian approximation q_L as the Laplace approximation, even if this phrase is sometimes used in Statistics for higher-order approximations, as discussed in the next Section. We defer to Section 2.3.5 the discussion of the advantages and drawbacks of this approximation scheme.

2.3.2 Improved Laplace, connection with INLA

Consider the marginal distributions $p(\beta_j|\mathcal{D}) = \int p(\boldsymbol{\beta}|\mathcal{D})\mathrm{d}\boldsymbol{\beta}_{-j}$ for each component β_j of $\boldsymbol{\beta}$, where $\boldsymbol{\beta}_{-j}$ is $\boldsymbol{\beta}$ minus β_j. A first approximation may be obtained by simply computing the marginals of the Laplace approximation q_L. An improved (but more expensive) approximation may be obtained from:

$$p(\beta_j|\mathcal{D}) \propto \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\boldsymbol{\beta}_{-j}|\beta_j,\mathcal{D})}$$

which suggests to choose a fine grid of β_j values (deduced for instance from $q_L(\boldsymbol{\beta})$), and for each β_j value, compute a Laplace approximation of $p(\boldsymbol{\beta}_{-j}|\beta_j,\mathcal{D})$, by computing the mode $\hat{\boldsymbol{\beta}}_{-j}(\beta_j)$ and the Hessian $\hat{\boldsymbol{H}}(\beta_j)$ of $\log p(\boldsymbol{\beta}_{-j}|\beta_j,\mathcal{D})$, and then approximate (up to a constant)

$$p(\beta_j|\mathcal{D}) \approx q_{IL}(\beta_j) \propto \frac{p\left(\hat{\boldsymbol{\beta}}(\beta_j)\right)p(\mathcal{D}|\hat{\boldsymbol{\beta}}(\beta_j))}{\left|\hat{H}(\beta_j)\right|^{1/2}}$$

where $\hat{\boldsymbol{\beta}}(\beta_j)$ is the vector obtained by inserting β_i at position i in $\hat{\boldsymbol{\beta}}_{-j}(\beta_j)$, and IL stands for "Improved Laplace". One may also deduce posterior expectations of functions of β_j in this way. See also Tierney and Kadane [1986], Tierney et al. [1989] for higher order approximations for posterior expectations.

We note in passing the connection to the INLA scheme of Rue et al. [2009]. INLA applies to posteriors $p(\boldsymbol{\theta},\boldsymbol{x}|\mathcal{D})$ where $\boldsymbol{x}$ is a latent variable such that $p(\boldsymbol{x}|\boldsymbol{\theta},\mathcal{D})$ is close to a Gaussian, and $\boldsymbol{\theta}$ is a low-dimensional hyper-parameter. It constructs a grid of $\boldsymbol{\theta}-$values, and for each grid point $\boldsymbol{\theta}_j$, it computes an improve Laplace approximation of the marginals of $p(\boldsymbol{x}|\boldsymbol{\theta}_j,\mathcal{D})$. In our context, $\boldsymbol{\beta}$ may be identified to $\boldsymbol{x}$, $\boldsymbol{\theta}$ to an empty set, and INLA reduces to the improved Laplace approximation described above.

2.3.3 The EM algorithm of Gelman et al. [2008] (Cauchy prior)

Gelman et al. [2008] recommend against the Laplace approximation for a Student prior (of which our Cauchy prior is a special case), because, as explained in Section 2.2.3, the corresponding log-posterior is not guaranteed to be concave, and this might prevent Newton-Raphson to converge. In our simulations however, we found the Laplace approximation to work reasonably well for a Cauchy prior. We now briefly describe the alternative approximation scheme proposed by Gelman et al. [2008] for Student priors, which we call for convenience Laplace-EM.

Laplace-EM is based on the well-known representation of a Student distribution, $\beta_j|\sigma_j^2 \sim \mathrm{N}_1(0, \sigma_j^2)$, $\sigma_j^2 \sim \mathrm{Inv}-\mathrm{Gamma}(\nu/2, s_j\nu/2)$; take $\nu = 1$ to recover our Cauchy prior. Conditional on $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_p^2)$, the prior on $\boldsymbol{\beta}$ is Gaussian, hence, for a fixed $\boldsymbol{\sigma}^2$ one may implement Newton-Raphson to maximise the log-density of $p(\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \mathcal{D})$, and deduce a Laplace (Gaussian) approximation of the same distribution.

Laplace-EM is an approximate EM [Expectation Maximisation, Dempster et al., 1977] algorithm, which aims at maximising in $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_p^2)$ the marginal posterior distribution $p(\boldsymbol{\sigma}^2|\mathcal{D}) = \int p(\boldsymbol{\sigma}^2, \boldsymbol{\beta}|\mathcal{D})\,\mathrm{d}\boldsymbol{\beta}$. Each iteration involves an expectation with respect to the intractable conditional distribution $p(\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \mathcal{D})$, which is Laplace approximated, using a single Newton-Raphson iteration. When this approximate EM algorithm has converged to some value $\boldsymbol{\sigma}_\star^2$, one more Newton-Raphson iteration is performed to compute a final Laplace approximation of $p(\boldsymbol{\beta}|\boldsymbol{\sigma}_\star^2, \mathcal{D})$, which is then reported as a Gaussian approximation to the posterior. We refer the readers to Gelman et al. [2008] for more details on Laplace-EM.

2.3.4 Expectation-Propagation

Like Laplace, Expectation Propagation [EP, Minka, 2001b] generates a Gaussian approximation of the posterior, but it is based on different ideas. The consensus in machine learning seems to be that EP provides a better approximation than Laplace [e.g. Nickisch and Rasmussen, 2008]; the intuition being that Laplace is 'too local' (i.e. it fitted so at to match closely the posterior around the mode), while EP is able to provide a global approximation to the posterior.

Starting from the decomposition of the posterior as product of $(n_\mathcal{D} + 1)$ factors:

$$p(\boldsymbol{\beta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \prod_{i=0}^{n_\mathcal{D}} l_i(\boldsymbol{\beta}), \quad l_i(\boldsymbol{\beta}) = F(y_i \boldsymbol{\beta}^T \boldsymbol{x}_i) \text{ for } i \geq 1,$$

and l_0 is the prior, $l_0(\boldsymbol{\beta}) = p(\boldsymbol{\beta})$, EP computes iteratively a parametric approxi-

mation of the posterior with the same structure

$$q_{\mathrm{EP}}(\boldsymbol{\beta}) = \prod_{i=0}^{n_{\mathcal{D}}} \frac{1}{Z_i} q_i(\boldsymbol{\beta}). \tag{2.4}$$

Taking q_i to be an unnormalised Gaussian densities written in natural exponential form

$$q_i(\boldsymbol{\beta}) = \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T \boldsymbol{Q}_i \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{r}_i\right\},$$

one obtains for q_{EP} a Gaussian with natural parameters $\boldsymbol{Q} = \sum_{i=0}^{n} \boldsymbol{Q}_i$ and $\boldsymbol{r}_i = \sum_{i=0}^{n} \boldsymbol{r}_i$; note that the more standard parametrisation of Gaussians may be recovered by taking

$$\boldsymbol{\Sigma} = \boldsymbol{Q}^{-1}, \quad \boldsymbol{\mu} = \boldsymbol{Q}^{-1}\boldsymbol{r}.$$

Other exponential families could be considered for q and the q_i's, see e.g. Seeger [2005b], but Gaussian approximations seems the most natural choice here.

An EP iteration consists in updating one factor q_i, or equivalently $(Z_i, \boldsymbol{Q}_i, \boldsymbol{r}_i)$, while keeping the other factors as fixed, by moment matching between the hybrid distribution

$$h(\boldsymbol{\beta}) \propto l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta})$$

and the global approximation q defined in (2.4): compute

$$\begin{aligned}
Z_h &= \int l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta} \\
\boldsymbol{\mu}_h &= \frac{1}{Z_h} \int \boldsymbol{\beta} l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta} \\
\boldsymbol{\Sigma}_h &= \frac{1}{Z_h} \int \boldsymbol{\beta}\boldsymbol{\beta}^T l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\beta}
\end{aligned}$$

and set

$$\boldsymbol{Q}_i = \boldsymbol{\Sigma}_h^{-1} - \boldsymbol{Q}_{-i}, \quad \boldsymbol{r}_i = \boldsymbol{\Sigma}_h^{-1}\boldsymbol{\mu}_h - \boldsymbol{r}_{-i}, \quad \log Z_i = \log Z_h - \Psi(\boldsymbol{r}, \boldsymbol{Q}) + \Psi(\boldsymbol{r}_{-i}, \boldsymbol{Q}_{-i})$$

where $\boldsymbol{r}_{-i} = \sum_{j \neq i} \boldsymbol{r}_j$, $\boldsymbol{Q}_{-i} = \sum_{j \neq i} \boldsymbol{Q}_j$, and $\psi(\boldsymbol{r}, \boldsymbol{Q})$ is the normalising constant of a Gaussian distribution with natural parameters $(\boldsymbol{r}, \boldsymbol{Q})$,

$$\psi(\boldsymbol{r}, \boldsymbol{Q}) = \int_{\mathbb{R}^p} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T \boldsymbol{Q} \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{r}\right\} \mathrm{d}\boldsymbol{\beta} = -\frac{1}{2}\log|\boldsymbol{Q}/2\pi| + \frac{1}{2}\boldsymbol{r}^T \boldsymbol{Q} \boldsymbol{r}.$$

In practice, EP proceeds by looping over sites, updating each one in turn until convergence is achieved.

To implement EP for binary regression models, two points must be addressed. First, how to compute the hybrid moments? For the probit model, these moments may be computed exactly, see the supplement to the paper, while for the other links function (such as logistic), numerical (one-dimensional) quadrature may be used. Second, how to deal with the prior? If the prior is Gaussian, one may simply set q_0 to the prior, and never update q_0 in the course of the algorithm. For a Cauchy prior, q_0 is simply treated as an extra site.

EP being a fairly recent method, it is currently lacking in terms of supporting theory, both in terms of algorithmic convergence (does it converge in a finite number of iterations?), and statistical convergence (does the resulting approximation converges in some sense to the true posterior distribution as $n_{\mathcal{D}} \to +\infty$?). On the other hand, there is mounting evidence that EP works very well in many problems; again see e.g. Nickisch and Rasmussen [e.g. 2008].

2.3.5 Discussion of the different approximation schemes

Laplace (and its variants) have complexity $\mathcal{O}(n_{\mathcal{D}} + p^3)$, while EP has complexity $\mathcal{O}(n_{\mathcal{D}} p^3)$. Incidentally, one sees that the number of covariates p is more critical than the number of instances $n_{\mathcal{D}}$ in determining how 'big' (how time-intensive to process) is a given dataset. This will be a recurring point in this chapter.

The p^3 term in both complexities is due to the $p \times p$ matrix operations performed by both algorithms; e.g. the Newton-Raphson update (2.2) requires solving a linear system of order p. EP requires to perform such p^3 operations at each site (i.e. for each single observation), hence the $\mathcal{O}(n_{\mathcal{D}} p^3)$ complexity, while Laplace perform such operations only once per iteration. EP is therefore expected to be more expensive than Laplace.

This remark may be mitigated as follows. First, one may modify EP so as to update the global approximation only at the end of each iteration (complete pass over the data). The resulting algorithm [Van Gerven et al., 2010] may be easily implemented on parallel hardware: simply distribute the $n_{\mathcal{D}}$ factors over the processors. Even without parallelisation, parallel EP requires only one single matrix inversion per iteration.

Second, the 'improved Laplace' approximation for the marginals described in Section 2.3.1 requires to perform quite a few basic Laplace approximations, so its speed advantage compared to standard EP essentially vanishes.

Points that remain in favour of Laplace is that it is simpler to implement than EP, and the resulting code is very generic: adapting to either a different prior, or a different link function (choice of F in 2.1), is simply a matter of writing a function that evaluates the corresponding function. We have seen that such an

adaptation requires more work in EP, although to be fair the general structure of the algorithm is not model-dependent. On the other hand, we shall see that EP is often more accurate, and works in more examples, than Laplace; this is especially the case for the Cauchy prior.

2.4 Exact methods

We now turn to sampling-based methods, which are 'exact', at least in the limit: one may make the approximation error as small as desired, by running the corresponding algorithm for long enough. We will see that all of these algorithms requires some form of calibration that requires prior knowledge on the shape of the posterior distribution. Since the approximation methods covered in the previous section are faster by orders of magnitude than sampling-based methods, we will assume that a Gaussian approximation $q(\boldsymbol{\beta})$ (say, obtained by Laplace or EP) has been computed in a preliminary step.

2.4.1 Our gold standard: Importance sampling

Let $q(\boldsymbol{\beta})$ denote a generic approximation of the posterior $p(\boldsymbol{\beta}|\mathcal{D})$. Importance sampling (IS) is based on the trivial identity

$$p(\mathcal{D}) = \int p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} = \int q(\boldsymbol{\beta})\frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})}\,\mathrm{d}\boldsymbol{\beta}$$

which leads to the following recipe: sample $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N \sim q$, then compute as an estimator of $p(\mathcal{D})$

$$Z_N = \frac{1}{N}\sum_{n=1}^{N} w(\boldsymbol{\beta}_n), \quad w(\boldsymbol{\beta}) := \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})}. \tag{2.5}$$

In addition, since

$$\int \varphi(\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathcal{D})\,\mathrm{d}\boldsymbol{\beta} = \frac{\int \varphi(\boldsymbol{\beta})q(\boldsymbol{\beta})w(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}}{\int q(\boldsymbol{\beta})w(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}}$$

one may approximate any posterior moment as

$$\varphi_N = \frac{\sum_{n=1}^{N} w(\boldsymbol{\beta}_n)\varphi(\boldsymbol{\beta}_n)}{\sum_{n=1}^{N} w(\boldsymbol{\beta}_n)}. \tag{2.6}$$

Approximating posterior marginals is also straightforward; one may instance use kernel density estimation on the weighted sample $(\boldsymbol{\beta}_n, w(\boldsymbol{\beta}_n))_{n=1}^{N}$.

Concerning the choice of q, we will restrict ourselves to the Gaussian approximations generated either from Laplace or EP algorithm. It is sometimes recommended to use a Student distribution instead, as a way to ensure that the variance of the above estimators is finite, but we did not observe any benefit for doing so in our simulations.

It is of course a bit provocative to call IS our gold standard, as it is sometimes perceived as an obsolete method. We would like to stress out however that IS is hard to beat relative to most of the criteria laid out in the introduction:

- because it is based on IID sampling, assessing the Monte Carlo error of the above estimators is trivial: e.g. the variance of Z_N may be estimated as N^{-1} times the empirical variance of the weights $w(\boldsymbol{\beta}_n)$. The auto-normalised estimator 2.6 has asymptotic variance

$$\mathbb{E}_q\left[w(\boldsymbol{\beta})^2\left\{\varphi(\boldsymbol{\beta})-\mu(\varphi)\right\}^2\right], \quad \mu(\varphi)=\int \varphi(\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathcal{D})\,\mathrm{d}\boldsymbol{\beta}$$

  which is also trivial to approximate from the simulated $\boldsymbol{\beta}_n$'s.

- Other advantages brought by IID sampling are: (a) importance sampling is easy to parallelize; and (b) importance sampling is amenable to QMC (Quasi-Monte Carlo) integration, as explained in the following section.

- Importance sampling offers an approximation of the marginal likelihood $p(\mathcal{D})$ at no extra cost.

- Code is simple and generic.

Of course, what remains to determine is whether importance sampling does well relative to our main criterion, i.e. error versus CPU trade-off. We do know that IS suffers from a curse of dimensionality: take both q and and the target density π to be the density of IID distributions: $q(\boldsymbol{\beta})=\prod_{j=1}^p q_1(\beta_j)$, $\pi(\boldsymbol{\beta})=\prod_{j=1}^p \pi_1(\beta_j)$; then it is easy to see that the variance of the weights grows exponentially with p. Thus we expect IS to collapse when p is too large; meaning that a large proportion of the $\boldsymbol{\beta}_n$ gets a negligible weight. On the other hand, for small to moderate dimensions, we will observe surprising good results; see Section 2.5. We will also present below a SMC algorithm that automatically reduces to IS when IS performs well, while doing something more elaborate in more difficult scenarios.

The standard way to assess the weight degeneracy is to compute the effective sample size [Kong et al., 1994],

$$\mathrm{ESS}=\frac{\left\{\sum_{n=1}^N w(\boldsymbol{\beta}_n)\right\}^2}{\sum_{n=1}^N w(\boldsymbol{\beta}_n)^2}\in[1,N],$$

which roughly approximates how many simulations from the target distribution would be required to produce the same level of error. In our simulations, we will compute instead the efficiency factor EF, which is simply the ratio EF = ESS/N.

2.4.2 Improving importance sampling by Quasi-Monte Carlo

Quasi-Monte Carlo may be seen as an elaborate variance reduction technique: starting from the Monte Carlo estimators Z_N and φ_N, see (2.5) and (2.6), one may re-express the simulated vectors as functions of uniform variates $\boldsymbol{u}_n$ in $[0,1]^d$; for instance:

$$\boldsymbol{\beta}_n = \boldsymbol{\mu} + \boldsymbol{C}\boldsymbol{\zeta}_n, \quad \boldsymbol{\zeta}_n = \boldsymbol{\Phi}^{-1}(\boldsymbol{u}_n)$$

where $\boldsymbol{\Phi}^{-1}$ is Φ^{-1}, the $N(0,1)$ inverse CDF, applied component-wise. Then, one replaces the N vectors $\boldsymbol{u}_n$ by a low-discrepancy sequence; that is a sequence of N vectors that spread more evenly over $[0,1]^d$; e.g. a Halton or a Sobol' sequence. Under appropriate conditions, QMC error converges at rate $\mathcal{O}(N^{-1+\epsilon})$, for any $\epsilon > 0$, to be compared with the standard Monte Carlo rate $\mathcal{O}_P(N^{-1/2})$. We refer to Lemieux [2009] for more background on QMC, as well as how to construct QMC sequences.

Oddly enough, the possibility to use QMC in conjunction with importance sampling is very rarely mentioned in the literature; see however Hörmann and Leydold [2005]. More generally, QMC seems often overlooked in Statistics. We shall see however that this simple IS-QMC strategy often performs very well.

One drawback of IS-QMC is that we lose the ability to evaluate the approximation error in a simple manner. A partial remedy is to use randomised Quasi-Monte Carlo (RQMC), that is, the $\boldsymbol{u}_n$ are generated in such a way that (a) with probability one, $\boldsymbol{u}_{1:N}$ is a QMC point set; and (b) each vector $\boldsymbol{u}_n$ is marginally sampled from $[0,1]^d$. Then QMC estimators that are empirical averages, such as $Z_N = N^{-1}\sum_{n=1}^{N} w(\boldsymbol{\beta}_n)$ become unbiased estimators, and their error may be assessed through the empirical variance over repeated runs. Technically, estimators that are ratios of QMC averages, such as φ_N, are not unbiased, but for all practical purposes their bias is small enough that assessing error through empirical variances over repeated runs remains a reasonable approach.

2.4.3 MCMC

The general principle of MCMC (Markov chain Monte Carlo) is to simulate a Markov chain that leaves invariant the posterior distribution $p(\boldsymbol{\beta}|\mathcal{D})$; see Robert and Casella [2004a] for a general overview. Often mentioned drawbacks of MCMC simulation are (a) the difficulty to parallelize such algorithms (although see e.g. Jacob et al.,

2011 for an attempt at this problem); (b) the need to specify a good starting point for the chain (or alternatively to determine the burn-in period, that is, the length of the initial part of the chain that should be discarded) and (c) the difficulty to assess the convergence of the chain (that is, to determine if the distribution of $\boldsymbol{\beta}_t$ at iteration t is sufficiently close to the invariant distribution $p(\boldsymbol{\beta}|\mathcal{D})$).

To be fair, these problems are not so critical for binary regression models. Regarding (b), one may simply start the chain from the posterior mode, or from a draw of one of the Gaussian approximations covered in the previous section. Regarding (c) for most standard datasets, MCMC converges reasonably fast, and convergence is easy to assess visually. The main issue in practice is that MCMC generates correlated random variables, and these correlations inflate the Monte Carlo variance.

Gibbs sampling

Consider the following data-augmentation formulation of binary regression:

$$\begin{aligned} z_i &= \boldsymbol{\beta}^T \boldsymbol{x}_i + \epsilon_i \\ y_i &= \mathrm{sgn}(z_i) \end{aligned}$$

where $\boldsymbol{z} = (z_1, \ldots, z_{n_\mathcal{D}})^T$ is a vector of latent variables, and assume for a start that $\epsilon_i \sim N(0,1)$ (probit regression). One recognises $p(\boldsymbol{\beta}|\boldsymbol{z}, \mathcal{D})$ as the posterior of a linear regression model, which is tractable (for an appropriate prior). This suggests to sample from $p(\boldsymbol{\beta}, \boldsymbol{z}|\mathcal{D})$ using Gibbs sampling [Albert and Chib, 1993]: i.e. iterate the two following steps: (a) sample from $\boldsymbol{z}|\boldsymbol{\beta}, \mathcal{D}$; and (b) sample from $\boldsymbol{\beta}|\boldsymbol{z}, \mathcal{D}$.

For (a), the z_i's are conditionally independent, and follows a truncated Gaussian distribution

$$p(z_i|\boldsymbol{\beta}, \mathcal{D}) \propto \mathrm{N}_1\left(z_i; \boldsymbol{\beta}^T \boldsymbol{x}_i, 1\right) \mathbb{1}\left\{z_i y_i > 0\right\}$$

which is easy to sample from [Chopin, 2011b]. For Step (b) and a Gaussian prior $\mathrm{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{prior}})$, one has, thanks to standard conjugacy properties:

$$\boldsymbol{\beta}|\boldsymbol{z}, \mathcal{D} \sim \mathrm{N}_p\left(\boldsymbol{\mu}_{\mathrm{post}}(\boldsymbol{z}), \boldsymbol{\Sigma}_{\mathrm{post}}\right), \quad \boldsymbol{\Sigma}_{\mathrm{post}}^{-1} = \boldsymbol{\Sigma}_{\mathrm{prior}}^{-1} + \boldsymbol{x}\boldsymbol{x}^T, \quad \boldsymbol{\mu}_{\mathrm{post}}(\boldsymbol{z}) = \boldsymbol{\Sigma}_{\mathrm{post}}^{-1} \boldsymbol{x}\boldsymbol{z}$$

where $\boldsymbol{x}$ is the $n \times p$ matrix obtained by stacking the $\boldsymbol{x}_i^T$. Note that $\boldsymbol{\Sigma}_{\mathrm{post}}$ and its inverse need to be computed only once, hence the complexity of a Gibbs iteration is $\mathcal{O}(p^2)$, not $\mathcal{O}(p^3)$.

The main drawback of Gibbs sampling is that it is particularly not generic: its implementation depends very strongly on the prior and the model. Sticking to the probit case, switching to another prior requires deriving a new way to update $\boldsymbol{\beta}|\boldsymbol{z}, \mathcal{D}$. For instance, for a prior which is a product of Students with scales σ_j

(e.g. our Cauchy prior), one may add extra latent variables, by resorting to the well-known representation: $\beta_j|s_j \sim \mathrm{N}_1(0, \nu\sigma_j^2/s_j)$, $s_j \sim \mathrm{Chi}^2(\nu)$; with $\nu = 1$ for our Cauchy prior. Then the algorithm have three steps: (a) an update of the z_i's, exactly as above; (b) an update of $\boldsymbol{\beta}$, as above but with $\boldsymbol{\Sigma}_{\text{prior}}$ replaced by the diagonal matrix with elements $\nu\sigma_j^2/s_j$, $j = 1, \ldots, p$; and (c) an (independent) update of the p latent variables s_j, with $s_j|\boldsymbol{\beta}, \boldsymbol{z}, \mathcal{D} \sim \text{Gamma}\left((1+\nu)/2, \left(1 + \nu\beta_j^2/\sigma_j^2\right)/2\right)$. The complexity of Step (b) is now $\mathcal{O}(p^3)$, since $\boldsymbol{\Sigma}_{\text{prior}}$ and $\boldsymbol{\Sigma}_{\text{post}}$ must be recomputed at each iteration.

Of course, considering yet another type of prior would require deriving another strategy for sampling $\boldsymbol{\beta}$. Then if one turns to logistic regression, things get rather complicated. In fact, deriving an efficient Gibbs sampler for logistic regression is a topic of current research; see Frühwirth-Schnatter and Frühwirth [2009]; Gramacy and Polson [2012]; Holmes and Held [2006]; Polson et al. [2013]. In a nutshell, the two first papers use the same data augmentation as above, but with $\epsilon_i \sim$ Logistic(1) written as a certain mixture of Gaussians (infinite for the first paper, finite but approximate for the second paper), while Polson et al. [2013] use instead a representation of a logistic likelihood as an infinite mixture of Gaussians, with a Polya-Gamma as the mixing distribution. Each representation leads to introducing extra latent variables, and discussing how to sample their conditional distributions.

Since their implementation is so model-dependent, the main justification for Gibbs samplers should be their greater performance relative to more generic algorithms. We will investigate if this is indeed the case in our numerical section.

Hastings-Metropolis

Hastings-Metropolis consists in iterating the step described as Algorithm 6. Much like importance sampling, Hastings-Metropolis is both simple and generic, that is, up to the choice of the proposal kernel $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta})$ (the distribution of the proposed point $\boldsymbol{\beta}^\star$, given the current point $\boldsymbol{\beta}$). A naive approach is to take $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta})$ independent of $\boldsymbol{\beta}$, $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta}) = q(\boldsymbol{\beta}^\star)$, where q is some approximation of the posterior. In practice, this usually does not work better than importance sampling based on the same proposal, hence this strategy is hardly used.

A more usual strategy is to set the proposal kernel to a random walk: $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta}) = \mathrm{N}_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\text{prop}})$. It is well known that the choice of $\boldsymbol{\Sigma}_{\text{prop}}$ is critical for good performance. For instance, in the univariate case, if $\boldsymbol{\Sigma}_{\text{prop}}$ is too small, the chain moves slowly, while if too large, proposed moves are rarely accepted.

A result from the optimal scaling literature [e.g. Roberts and Rosenthal, 2001] is that, for a $\mathrm{N}_p(\mathbf{0}, \boldsymbol{I}_p)$ target, $\boldsymbol{\Sigma}_{\text{prop}} = (\lambda^2/p)\boldsymbol{I}_p$ with $\lambda = 2.38$ is asymptotically optimal, in the sense that as $p \to \infty$, this choice leads to the fastest exploration. Since the posterior of a binary regression model is reasonably close to a Gaussian,

Algorithm 6 Hastings-Metropolis iteration

Input $\boldsymbol{\beta}$

Output $\boldsymbol{\beta}'$

1 Sample $\boldsymbol{\beta}^\star \sim \kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta})$.

2 With probability $1 \wedge r$,

$$r = \frac{p(\boldsymbol{\beta}^\star)p(\mathcal{D}|\boldsymbol{\beta}^\star)\kappa(\boldsymbol{\beta}|\boldsymbol{\beta}^\star)}{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta})},$$

set $\boldsymbol{\beta}' = \boldsymbol{\beta}^\star$; otherwise set $\boldsymbol{\beta}' = \boldsymbol{\beta}$.

we adapt this result by taking $\boldsymbol{\Sigma}_{\text{prop}} = (\lambda^2/p)\boldsymbol{\Sigma}_q$ in our simulations, where $\boldsymbol{\Sigma}_q$ is the covariance matrix of a (Laplace or EP) Gaussian approximation of the posterior. This strategy seems validated by the fact we obtain acceptance rates close to the optimal rate, as given by Roberts and Rosenthal [2001].

The bad news behind this optimality result is that the chain requires $\mathcal{O}(p)$ steps to move a $\mathcal{O}(1)$ distance. Thus random walk exploration tends to become slow for large p. This is usually cited as the main motivation to develop more elaborate MCMC strategies, such as HMC, which we cover in the following section.

HMC

Hamiltonian Monte Carlo (HMC, also known as Hybrid Monte Carlo, Duane et al., 1987) is a new type of MCMC algorithm, where one is able to perform several steps in the parameter space before determining if the new position is accepted or not. Consequently, HMC is able to make much bigger jumps in the parameter space than standard Metropolis algorithms. See Neal [2010b] for an excellent introduction.

Consider the pair $(\boldsymbol{\beta}, \boldsymbol{\alpha})$, where $\boldsymbol{\beta} \sim p(\boldsymbol{\beta}|\mathcal{D})$, and $\boldsymbol{\alpha} \sim N_p(0, M^{-1})$, thus with joint un-normalised density $\exp\{-H(\boldsymbol{\beta}, \boldsymbol{\alpha})\}$, with

$$H(\boldsymbol{\beta}, \boldsymbol{\alpha}) = E(\boldsymbol{\beta}) + \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}, \quad E(\boldsymbol{\beta}) = -\log\{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\}.$$

The physical interpretation of HMC is that of a particle at position $\boldsymbol{\beta}$, with velocity $\boldsymbol{\alpha}$, potential energy $E(\boldsymbol{\beta})$, kinetic energy $\frac{1}{2}\boldsymbol{\alpha}^T M \boldsymbol{\alpha}$, for some mass matrix M, and therefore total energy given by $H(\boldsymbol{\beta}, \boldsymbol{\alpha})$. The particle is expected to follow a trajectory such that $H(\boldsymbol{\beta}, \boldsymbol{\alpha})$ remains constant over time.

In practice, HMC proceeds as follows: first, sample a new velocity vector, $\boldsymbol{\alpha} \sim N_p(0, M^{-1})$. Second, move the particle while keeping the Hamiltonian H

constant; in practice, discretisation must be used, so L steps of step-size ϵ are performed through leap-frop steps; see Algorithm 7 which describes one such step. Third, the new position, obtained after L leap-frog steps is accepted or rejected according to probability $1 \wedge \exp\{H(\boldsymbol{\beta}, \boldsymbol{\alpha}) - H(\boldsymbol{\beta}^\star, \boldsymbol{\alpha}^\star)\}$; see Algorithm 8 for a summary. The validity of the algorithm relies on the fact that a leap-frog step is "volume preserving"; that is, the deterministic transformation $(\boldsymbol{\beta}, \boldsymbol{\alpha}) \to (\boldsymbol{\beta}_1, \boldsymbol{\alpha}_1)$ has Jacobian one. This is why the acceptance probability admits this simple expression.

Algorithm 7 Leap-frog step

Input $(\boldsymbol{\beta}, \boldsymbol{\alpha})$

Output $(\boldsymbol{\beta}_1, \boldsymbol{\alpha}_1)$

1 $\boldsymbol{\alpha}_{1/2} \leftarrow \boldsymbol{\alpha} - \frac{\epsilon}{2}\nabla_{\boldsymbol{\beta}} E(\boldsymbol{\beta})$

2 $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\beta} + \epsilon \boldsymbol{\alpha}_{1/2}$

3 $\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{\alpha}_{1/2} - \frac{\epsilon}{2}\nabla_{\boldsymbol{\beta}} E(\boldsymbol{\beta}_1)$

Algorithm 8 HMC iteration

Input $\boldsymbol{\beta}$

Output $\boldsymbol{\beta}'$

1 Sample momentum $\boldsymbol{\alpha} \sim \mathrm{N}_p(0, \boldsymbol{M})$.

2 Perform L leap-frog steps (see Algorithm 7), starting from $(\boldsymbol{\beta}, \boldsymbol{\alpha})$; call $(\boldsymbol{\beta}^\star, \boldsymbol{\alpha}^\star)$ the final position.

3 With probability $1 \wedge r$,

$$r = \exp\{H(\boldsymbol{\beta}, \boldsymbol{\alpha}) - H(\boldsymbol{\beta}^\star, \boldsymbol{\alpha}^\star)\}$$

set $\boldsymbol{\beta}' = \boldsymbol{\beta}^\star$; otherwise set $\boldsymbol{\beta}' = \boldsymbol{\beta}$.

The tuning parameters of HMC are $\boldsymbol{M}$ (the mass matrix), L (number of leap-frog steps), and ϵ (the stepsize). For $\boldsymbol{M}$, we follow Neal [2010b]'s recommendation and take $\boldsymbol{M}^{-1} = \boldsymbol{\Sigma}_q$, an approximation of the posterior variance (again obtained from either Laplace or EP). This is equivalent to rescaling the posterior so as to

have a covariance matrix close to identity. In this way, we avoid the bad mixing typically incurred by strong correlations between components.

The difficulty to choose L and ϵ seems to be the main drawback of HMC. The performance of HMC seems very sensitive to these tuning parameters, yet clear guidelines on how to choose them seem currently lacking. A popular approach is to fix $L\epsilon$ to some value, and to use vanishing adaptation [Andrieu and Thoms, 2008] to adapt ϵ so as to target acceptance rate of 0.65 (the optimal rate according to the formal study of HMC by Beskos et al., 2013): i.e. at iteration t, take $\epsilon = \epsilon_t$, with $\epsilon_t = \epsilon_{t-1} - \eta_t(R_t - 0.65)$, $\eta_t = t^{-\kappa}$, $\kappa \in (1/2, 1)$ and R_t the acceptance rate up to iteration t. The rationale for fixing $L\epsilon$ is that quantity may be interpreted as a 'simulation length', i.e. how much distance one moves at each step; if too small, the algorithm may exhibit random walk behaviour, while if too large, it may move a long distance before coming back close to its starting point. Since the spread of is already taken into account through $\boldsymbol{M}^{-1} = \boldsymbol{\Sigma}_q$, we took $\epsilon L = 1$ in our simulations.

NUTS and other variants of HMC

Girolami and Calderhead [2011] proposed an interesting variation of HMC, where the mass matrix $\boldsymbol{M}$ is allowed to depends on $\boldsymbol{\beta}$; e.g. $\boldsymbol{M}(\boldsymbol{\beta})$ is set to the Fisher information of the model. This allows the corresponding algorithm, called RHMC (Riemanian HMC), to adapt locally to the geometry of the target distribution. The main drawback of RHMC is that each iteration involves computing derivatives of $M(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, which is very expensive, especially if p is large. For binary regression, we found RMHC to be too expensive relative to plain HMC, even when taking into account the better exploration brought by RHMC. This might be related to the fact that the posterior of a binary regression model is rather Gaussian-like and thus may not require such a local adaptation of the sampler.

We now focus on NUTS [No U-Turn sampler, Hoffman and Gelman, 2013], a variant of HMC which does not require to specify a priori L, the number of leap-frog steps. Instead, NUTS aims at keeping on doing such steps until the trajectory starts to loop back to its initial position. Of course, the difficulty in this exercise is to preserve the time reversibility of the simulated Markov chain. To that effect, NUTS constructs iteratively a binary tree whose leaves correspond to different velocity-position pairs $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ obtained after a certain number of leap-frog steps. The tree starts with two leaves, one at the current velocity-position pair, and another leaf that corresponds to one leap-frop step, either in the forward or backward direction (i.e. by reversing the sign of velocity); then it iteratively doubles the number of leaves, by taking twice more leap frog steps, again either in the forward or backward direction. The tree stops growing when at least one leaf corresponds to a "U-turn"; then NUTS chooses randomly one leaf, among those

leaves that would have generated the current position with the same binary tree mechanism; in this way reversibility is preserved. Finally NUTS moves the new position that corresponds to the chosen leaf.

We refer the readers to Hoffman and Gelman [2013] for a more precise description of NUTS. Given its complexity, implementing directly NUTS seems to require more efforts than the other algorithms covered in this chapter. Fortunately, the STAN package (`http://mc-stan.org/`) provides a C++ implementation of NUTS which is both efficient and user-friendly: the only required input is a description of the model in a probabilistic programming language similar to BUGS. In particular, STAN is able to automatically derive the log-likelihood and its gradient, and no tuning of any sort is required from the user. Thus, we will use STAN to assess NUTS in our numerical comparisons.

2.4.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a class of algorithms for approximating iteratively a sequence of distributions π_t, $t = 0, \ldots, T$, using importance sampling, resampling, and MCMC steps. We focus here on the non-sequential use of SMC [Chopin, 2002b; Del Moral et al., 2006a; Neal, 2001b], where one is only interested in approximating the final distribution π_T (in our case, set to the posterior $p(\boldsymbol{\beta}|\mathcal{D})$), and the previous π_t's are designed so as to allow for a smooth progression from some π_0, which is easy to sample from, to π_T.

At iteration t, SMC produces a set of weighted particles (simulations) $(\boldsymbol{\beta}_n, w_n)_{n=1}^N$ that approximates π_t, in the sense that

$$\frac{1}{\sum_{n=1}^N w_n} \sum_{n=1}^N w_n \varphi(\boldsymbol{\beta}_n) \to \mathbb{E}^{\pi_t} \left[\varphi(\boldsymbol{\beta})\right]$$

as $N \to +\infty$. At time 0, one samples $\boldsymbol{\beta}^n \sim \pi_0$, and set $w_n = 1$. To progress from π_{t-1} to π_t, one uses importance sampling: weights are multiplied by ratio $\pi_t(\boldsymbol{\beta}_n)/\pi_{t-1}(\boldsymbol{\beta}_n)$. When the variance of the weights gets too large (which indicates that too few particles contribute significantly to the current approximation), one resamples the particles: each particle gets reproduced O_n times, where $O_n \geq 0$ is random, and such that $\mathbb{E}(O_n) = N w_n / \sum_{m=1}^N w_m$, and $\sum_{n=1}^N O_n = N$ with probability one. In this way, particles with a low weights are likely to die, while particles with a large weight get reproduced many times. Finally, one may reintroduce diversity among the particles by applying one (or several) MCMC steps, using a MCMC kernel that leaves invariant the current distribution π_t.

We focus in this chapter on tempering SMC, where the sequence

$$\pi_t(\boldsymbol{\beta}) \propto q(\boldsymbol{\beta})^{1-\delta_t} \left\{p(\boldsymbol{\beta}) p(\mathcal{D}|\boldsymbol{\beta})\right\}^{\delta_t}$$

corresponds to a linear interpolation (on the log-scale) between some distribution $\pi_0 = q$, and $\pi_T(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathcal{D})$, our posterior. This is a convenient choice in our case, as we have at our disposal some good approximation q (either from Laplace or EP) of our posterior. A second advantage of tempering SMC is that one can automatically adapt the "temperature ladder" δ_t [Jasra et al., 2011a]. Algorithm 9 describes a tempering SMC algorithm based on such an adaptation scheme: at each iteration, the next distribution π_t is chosen so that the efficiency factor (defined in Section 2.4.1) of the importance sampling step from π_{t-1} to π_t equals a pre-defined level $\tau \in (0, 1)$; a default value is $\tau = 1/2$.

Algorithm 9 tempering SMC

Operations involving index n must be performed for all $n \in 1 : N$.

0 Sample $\boldsymbol{\beta}_n \sim q(\boldsymbol{\beta})$ and set $\underline{\delta} \leftarrow 0$.

1 Let, for $\delta \in [\underline{\delta}, 1]$,

$$\mathrm{EF}(\delta) = \frac{1}{N} \frac{\left\{\sum_{n=1}^N w_\gamma(\boldsymbol{\beta}_n)\right\}^2}{\left\{\sum_{n=1}^N w_\gamma(\boldsymbol{\beta}_n)^2\right\}}, \quad u_\delta(\boldsymbol{\beta}) = \left\{\frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})}\right\}^\delta.$$

If $\mathrm{EF}(1) \geq \tau$, stop and return $(\boldsymbol{\beta}_n, w_n)_{n=1:N}$ with $w_n = u_1(\boldsymbol{\beta}_n)$; otherwise, use the bisection method [Press et al., 2007, Chap. 9] to solve numerically in δ the equation $\mathrm{EF}(\gamma) = \tau$.

2 Resample according to normalised weights $W_n = w_n / \sum_{m=1}^N w_m$, with $w_n = u_\delta(\boldsymbol{\beta}_n)$; see the supplement for one such resampling algorithm.

3 Update the $\boldsymbol{\beta}_n$'s through m MCMC steps that leaves invariant $\pi_t(\boldsymbol{\beta})$, using e.g. Algorithm 6 with $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta}) = \mathrm{N}_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\mathrm{prop}})$, $\boldsymbol{\Sigma}_{\mathrm{prop}} = \lambda\hat{\boldsymbol{\Sigma}}$, where $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix of the resampled particles.

4 Set $\underline{\delta} \leftarrow \delta$. Go to Step 1.

Another part of Algorithm 9 which is easily amenable to automatic calibration is the MCMC step. We use a random walk Metropolis step, i.e. Algorithm 6 with proposal kernel $\kappa(\boldsymbol{\beta}^\star|\boldsymbol{\beta}) = \mathrm{N}_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\mathrm{prop}})$, but with $\boldsymbol{\Sigma}_{\mathrm{prop}}$ calibrated to the empirical variance of the particles $\hat{\boldsymbol{\Sigma}}$: $\boldsymbol{\Sigma}_{\mathrm{prop}} = \lambda\hat{\boldsymbol{\Sigma}}$, for some λ. Finally, one may also automatically calibrate the number m of MCMC steps, as in Chapter 3, but in our simulations we simply took $m = 3$.

In the end, one obtains essentially a black-box algorithm. In practice, we shall

often observe that, for simple datasets, our SMC algorithm automatically reduces to a single importance sampling step, because the efficiency factor of moving from the initial distribution q to the posterior is high enough. In that case, our SMC sampler performs exactly as standard importance sampling.

2.5 Numerical study

The point of this section is to compare numerically the different methods discussed in the previous sections, first on several datasets of standard size (that are representative of previous numerical studies), then in a second time on several bigger datasets.

We focus on the following quantities: the marginal likelihood of the data, $p(\mathcal{D})$, and the p marginal posterior distributions of the regression coefficients β_j. Regarding the latter, we follow Faes et al. [2011] in defining the 'marginal accuracy' of approximation q for component j to be

$$\mathrm{MA}_j = 1 - \frac{1}{2}\int_{-\infty}^{+\infty} |q(\beta_j) - p(\beta_j|\mathcal{D})| \, \mathrm{d}\beta_j.$$

This quantity lies in $[0, 1]$, and is scale-invariant. Since the true marginals $p(\beta_j|\mathcal{D})$ are not available, we will approximate them through a Gibbs sampler run for a very long time. To give some scale to this criterion, assume $q(\beta_j) = \mathrm{N}_1(\beta_j; \mu_1, \sigma^2)$, $p(\beta_j|\mathcal{D}) = \mathrm{N}_1(\beta_j; \mu_2, \sigma^2)$, then MA_j is $2\Phi(-\delta/2) \approx 1 - 0.4 \times \delta$ for $\delta = |\mu_1 - \mu_2|/\sigma$ small enough; e.g. 0.996 for $\delta \approx 0.01$, 0.96 for $\delta \approx 0.1$.

In our results, we will refer to the following four prior/model 'scenarios': Gaussian/probit, Gaussian/logit, Cauchy/probit, Cauchy/logit, where Gaussian and Cauchy refer to the two priors discussed in Section 2.2.1. All the algorithms have been implemented in C++, using the Armadillo and Boost libraries, and run on a standard desktop computer (except when explicitly stated). Results for NUTS were obtained by running STAN (`http://mc-stan.org/`) version 2.4.0.

2.5.1 Datasets of moderate size

Table 2.1 lists the 7 datasets considered in this section (obtained from the UCI machine learning repository, except Elections, which is available on the web page of Gelman and Hill [2006]'s book). These datasets are representative of the numerical studies found in the literature. In fact, it is a super-set of the real datasets considered in Girolami and Calderhead [2011], Shahbaba et al. [2011], Holmes and Held [2006] and also (up to one dataset with 5 covariates) Polson et al. [2013]. In each case, an intercept have been included; i.e. p is the number of predictors plus one.

Dataset	$n_{\mathcal{D}}$	p
Pima (Indian diabetes)	532	8
German (credit)	999	25
Heart (Statlog)	270	14
Breast (cancer)	683	10
Liver (Indian Liver patient)	579	11
Plasma (blood screening data)	32	3
Australian (credit)	690	15
Elections	2015	52

Table 2.1: Datasets of moderate size (from UCI repository, except Elections, from web-site of Gelman and Hill [2006]'s book): name (short and long version), number of instances $n_{\mathcal{D}}$, number of covariates p (including an intercept)

Fast Approximations

We compare the four approximation schemes described in Section 2.3: Laplace, Improved Laplace, Laplace EM, and EP. We concentrate on the Cauchy/logit scenario for two reasons: (i) Laplace EM requires a Student prior; and (ii) Cauchy/logit seems the most challenging scenario for EP, as (a) a Cauchy prior is more difficult to deal with than a Gaussian prior in EP ; and (b) contrary to the probit case, the site update requires some approximation; see Section 2.3.4 for more details.

Left panel of Fig. 2.1 plots the marginal accuracies of the four approximation schemes across all components and all datasets; Fig. 2.2 does the same, but separately for four selected datasets; results for the remaining datasets are available in the supplement.

EP seems to be the most accurate method on these datasets: marginal accuracy is about 0.99 across all components for EP, while marginal accuracy of the other approximation schemes tend to be lower, and may even drop to quite small values; see e.g. the German dataset, and the left tail in the left panel of Fig. 2.1.

EP also fared well in terms of CPU time: it was at most seven times as intensive as standard Laplace across the considered datasets, and about 10 to 20 times faster than Improved Laplace and Laplace EM. As expected (see Section 2.3.5), the largest running time for EP was observed for the dataset with the largest number of observations (German credit): i.e. 0.14s, while Laplace took 0.02s on the same data. Of course, the usual caveats apply regarding CPU time comparison, and how they may depend on the hardware, the implementation, and so on.

We also note in passing the disappointing performance of Laplace EM, which was supposed to replace standard Laplace when the prior is Student, but which

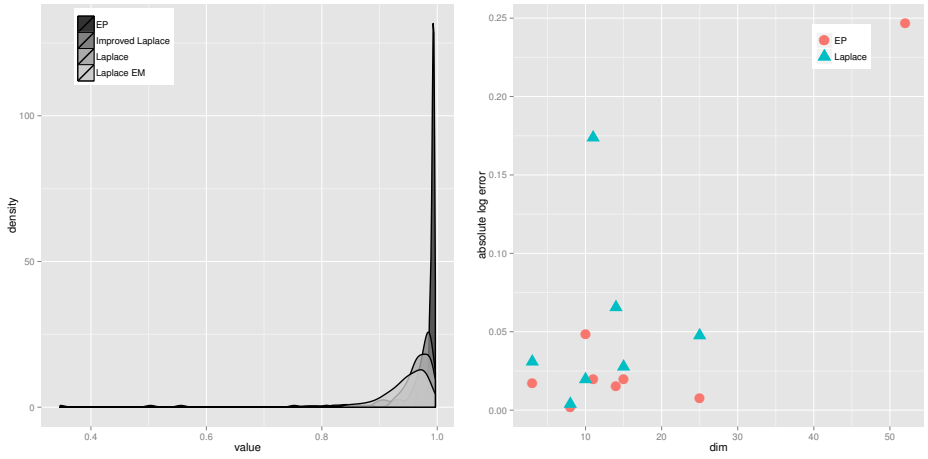


Figure 2.1: Comparison of approximation schemes across all datasets of moderate size: marginal accuracies (left), and absolute error for log-evidence versus the dimension p (right); $x-$axis range of the left plot determined by range of marginal accuracies (i.e. marginal accuracy may drop below 0.4 for e.g. Laplace-EM).

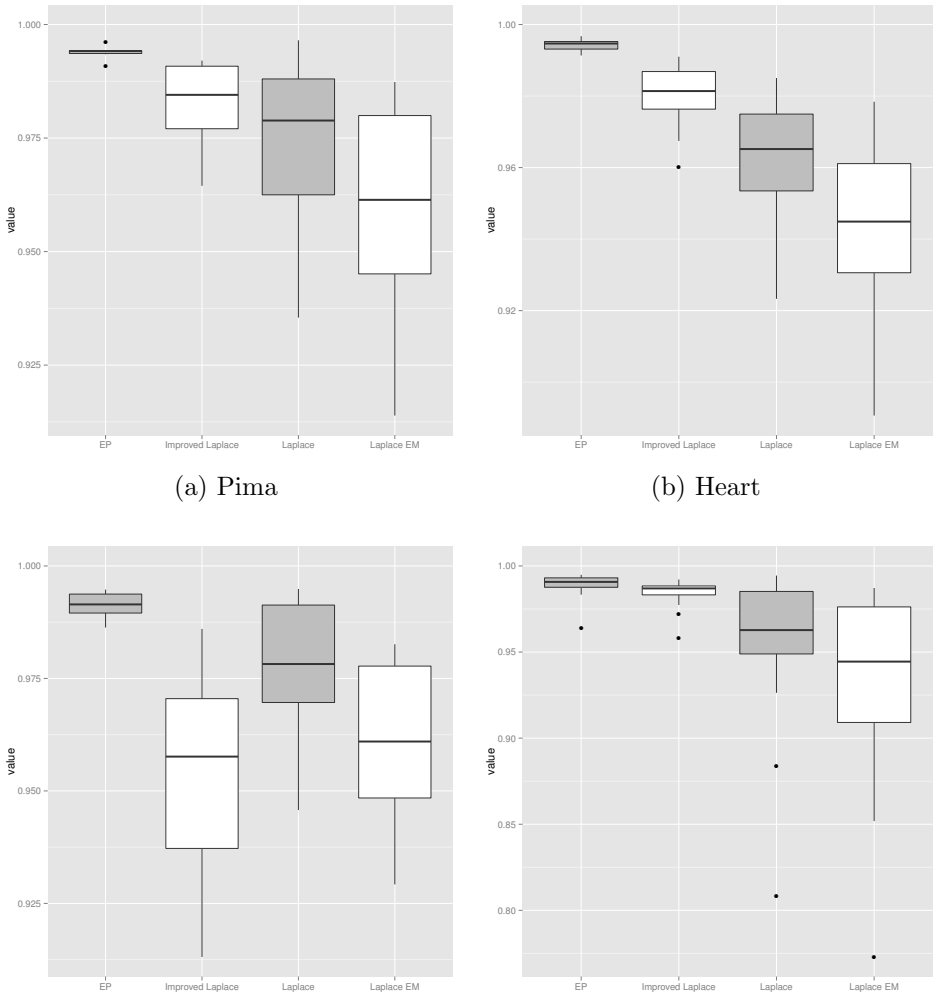
actually performs not as well as standard Laplace on these datasets.

We refer the reader to the supplement for similar results on the three other scenarios, which are consistent with those above. In addition, we also represent the approximation error of EP and Laplace for approximating the log-evidence in the right panel of Fig. 2.1. Again, EP is found to be more accurate than Laplace for most datasets (except for the Breast dataset).

To conclude, it seems that EP may be safely be used as a complete replacement of sampling-based methods on such datasets, as it produces nearly instant results, and the approximation error along all dimensions is essentially negligible.

Importance sampling, QMC

We now turn to importance sampling (IS), which we deemed our "gold standard" among sampling-based methods, because of its ease of use and other nice properties as discussed in Section 2.4.1. We use $N = 5 \times 10^5$ samples, and a Gaussian EP proposal. (Results with a Laplace proposal are roughly similar.) We consider first the Gaussian/probit scenario, because this is particularly favorable to Gibbs sampling; see next section. Table 2.2 reports for each dataset the efficiency factor of IS (as defined in Section 2.4.1), the CPU time and two other quantities discussed



(a) Pima (b) Heart

(c) Breast (d) German

Figure 2.2: Box-plots of marginal accuracies across the p dimensions, for the four approximation schemes, and four selected datasets; plots for remaining datasets are in the supplement. For the sake of readability, scale of $y-$axis varies across plots.

below.

	IS			IS-QMC	
Dataset	EF = ESS/N	CPU time	MT speed-up	MSE improv. (expectation)	MSE improv. (evidence)
Pima	99.5%	37.54 s	4.39	28.9	42.7
German	97.9%	79.65 s	4.51	13.2	8.2
Breast	82.9%	50.91 s	4.45	2.6	6.2
Heart	95.2%	22.34 s	4.53	8.8	9.3
Liver	74.2 %	35.93 s	4.76	7.6	11.3
Plasma	90.0%	2.32 s	4.28	2.2	4.4
Australian	95.6%	53.32 s	4.57	12	20.3
Elections	21.39%	139.48 s	3.87	617.9	3.53

Table 2.2: Performance of importance sampling (IS), and QMC importance sampling (IS-QMC), on all datasets, in Gaussian/probit scenario: efficiency factor (EF), CPU time (in seconds), speed gain when using multi-threading Intel hyper-threaded quad core CPU (Speed gain MT), and efficiency gain of QMC (see text).

We see that all these efficiency factors are all close to one, which means IS works almost as well as IID sampling would on such datasets. Further improvement may be obtained by using either parallelization, or QMC (Quasi-Monte Carlo, see Section 2.4.2). Table 2.2 reports the speed-up factor obtained when implementing multi-threading on our desktop computer which has a multi threading quad core CPU (hence 8 virtual cores). We also implemented IS on an Amazon EC2 instance with 32 virtual CPUs, and obtained speed-up factors about 20, and running times below $2s$.

Finally, Table 2.2 also reports the MSE improvement (i.e. MSE ratio of IS relative to IS-QMC) obtained by using QMC, or more precisely RQMC (randomised QMC), based on a scrambled Sobol' sequence [see e.g. Lemieux, 2009]. Specifically, the table reports the median MSE improvement for the p posterior expectations (first column), and the MSE improvement for the evidence (second column). The improvement brought by RQMC varies strongly across datasets.

The efficiency gains brought by parallelization and QMC may be combined, because the bulk of the computation (as reported by a profiler) is the N likelihood evaluations, which are trivial to parallelize.

It is already clear that other sampling-based methods do not really have a fighting chance on such datasets, but we shall compare them in the next section for

the sake of completeness. See also the supplement for results for other scenarios, which are very much in line with those above.

MCMC schemes

In order to compare the different sampling-based methods, we define the IRIS (Inefficiency Relative to Importance Sampling) criterion, for a given method M and a given posterior estimate, as follows:

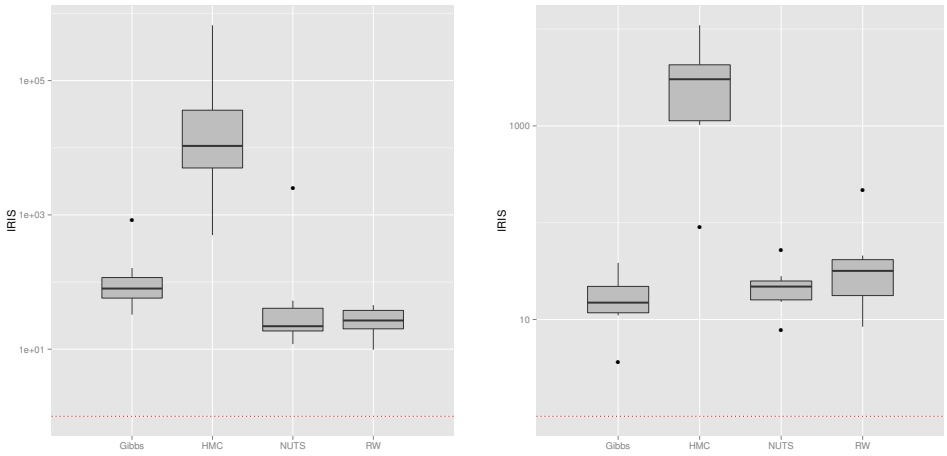
$$\frac{\text{MSE}_M}{MSE_{IS}} \times \frac{\text{CPU}_{IS}}{\text{CPU}_M}$$

where MSE_M (resp. MSE_{IS}) is the mean square error of the posterior estimate obtained from method M (resp. from importance sampling), and CPU_M the CPU time of method M (resp. importance sampling). The comparison is relative to importance sampling *without* parallelisation or quasi-Monte Carlo sampling. In terms of posterior estimates, we consider the expectation and variance of each posterior marginal $p(\beta_j|\mathcal{D})$. We observe that, in both cases, IRIS does not vary much across the p components, so we simply report the median of these p values. Fig 2.3 reports the median IRIS across all datasets. We refer the reader to Section 2.4.3 for how we tuned these MCMC algorithms.

The first observation is that all these MCMC schemes are significantly less efficient than importance sampling on such datasets. The source of inefficiency seems mostly due to the autocorrelations of the simulated chains (for Gibbs or random walk Metropolis), or, equivalently, the number of leap-frog steps performed at each iteration in HMC and NUTS. See the supplement for ACF's (Autocorrelation plots) to support this statement.

Second, HMC and NUTS do not perform significantly better than random-walk Metropolis. As already discussed, HMC-type algorithms are expected to outperform random walk algorithms as $p \to +\infty$. But the considered datasets seem too small to give evidence to this phenomenon, and should not be considered as reasonable benchmarks for HMC-type algorithms (not to mention again that these algorithms are significantly outperformed by IS on such datasets). We note in passing that it might be possible to get better performance for HMC by finely tuning the quantities ϵ and L on per dataset basis. We have already explained in the introduction why we think this is bad practice, and we also add at this stage that the fact HMC requires so much more effort to obtain good performance (relative to other MCMC samplers) is a clear drawback.

Regarding Gibbs sampling, it seems a bit astonishing that an algorithm specialised to probit regression is not able to perform better than more generic approach on such simple datasets. Recall that the Gaussian/probit case is particularly favourable to Gibbs, as explained in Section 2.4.3. See the supplement for



(a) Median IRIS for the p posterior expectations $\mathbb{E}[\beta_j|\mathcal{D}]$

(b) Median IRIS for the p posterior variances $\mathrm{Var}[\beta_j|\mathcal{D}]$

Figure 2.3: IRIS (Inefficiency relative to importance sampling) across all datasets for MCMC schemes and Gaussian/probit scenario; left (resp. right) panel shows median IRIS when estimating the p posterior expectations (resp. the p posterior variances).

Dataset	$n_{\mathcal{D}}$	p
Musk	476	95
Sonar	208	61
DNA	400	180

Table 2.3: Datasets of larger size (from UCI repository): name, number of instances $n_{\mathcal{D}}$, number of covariates p (including an intercept)

a comparison of MCMC schemes in other scenarios than Gaussian/probit; results are roughly similar, except that Gibbs is more significantly outperformed by other methods, as expected.

2.5.2 Bigger datasets

Finally, we turn our attention to the bigger datasets summarised by Table 2.3. These datasets not only have more covariates (than those of the previous section), but also stronger correlations between these covariates (especially Sonar and Musk). We consider the probit/Gaussian scenario.

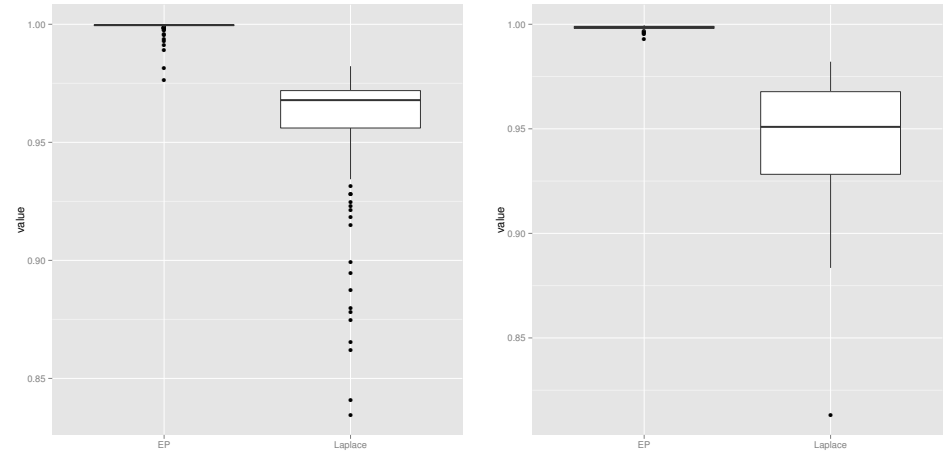
Regarding fast approximations, we observe again that EP performs very well, and better than Laplace; see Figure 2.5. It is only for DNA (180 covariates) that the EP approximation starts to suffer.

Regarding sampling-based methods, importance sampling may no longer be used as a reference, as the effective sample size collapses to a very small value for these datasets. We replace it by the tempering SMC algorithm described in Section 2.4.4. Moreover, we did not manage to calibrate HMC so as to obtain reasonable performance in this setting. Thus, among sampling-based algorithms, the four remaining contenders are: Gibbs sampling, NUTS, RWHM (random walk Hastings-Metropolis), and tempering SMC. Recall that the last two are calibrated with the approximation provided by EP.

Figure 2.5 reports the "effective sample size" of the output of these algorithms when run for the same fixed CPU time (corresponding to 5×10^5 iterations of RWHM), for the p posterior expectations (left panels), and the p posterior variances (right panels); here "effective sample size" is simply the posterior variance divided by the MSE of the estimate (across 50 independent runs of the same algorithm).

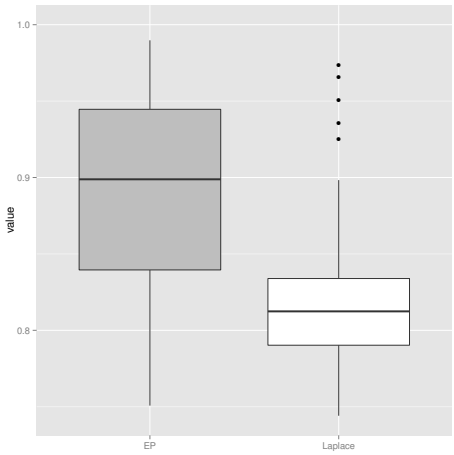
No algorithm seems to vastly outperform the others consistently across the three datasets. If anything, RWMH seems to show consistently best or second best performance.

Still, these results offer the following insights. Again, we see that Gibbs sam-



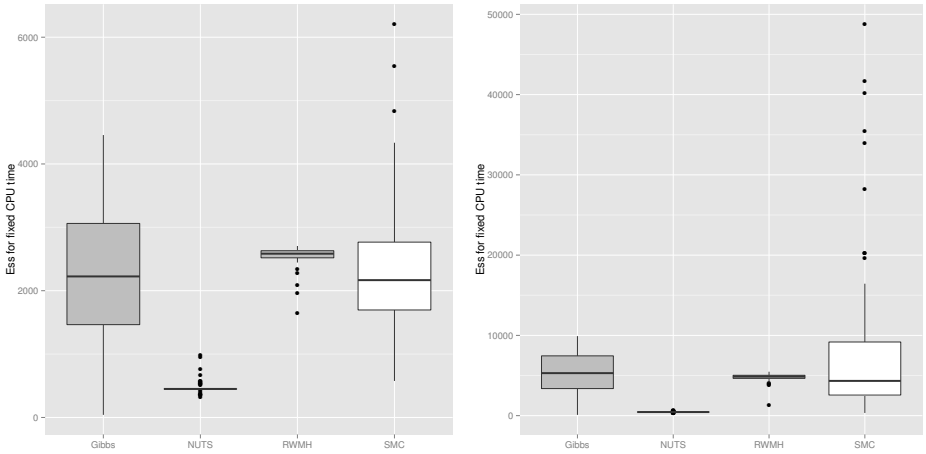
(a) Musk

(b) Sonar

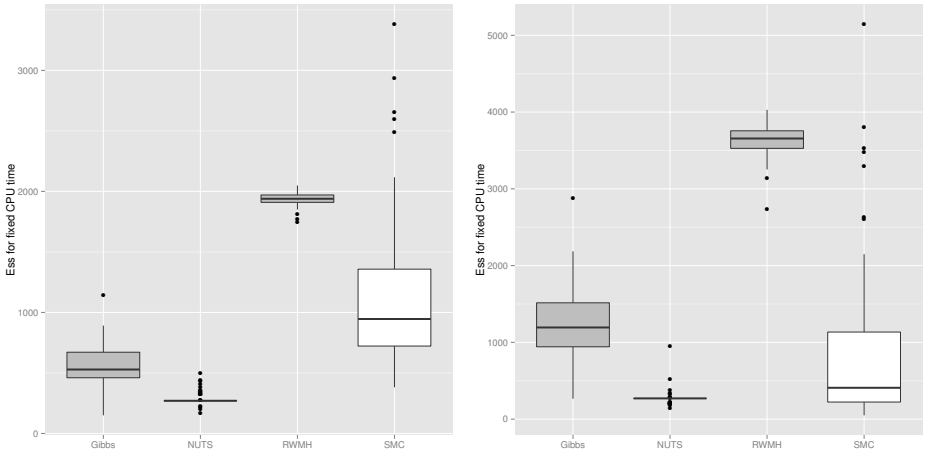


(c) DNA

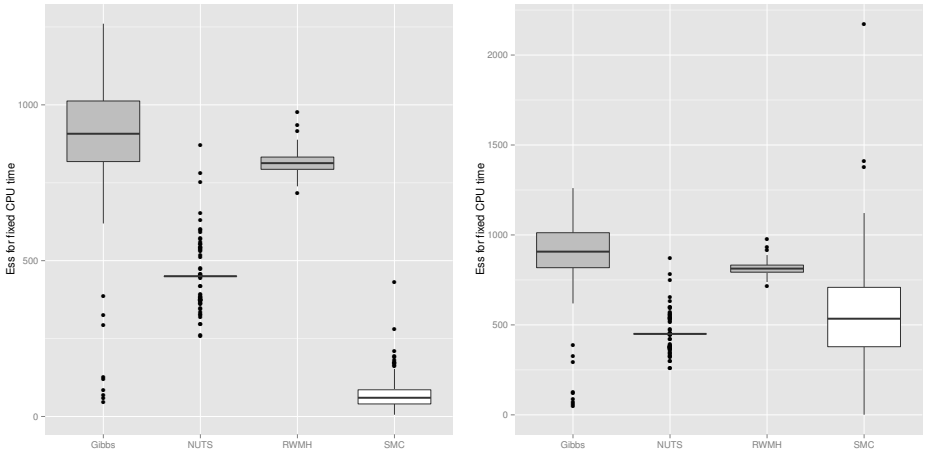
Figure 2.4: Marginal accuracies across the p dimensions of EP and Laplace, for datasets Musk, Sonar and DNA



(a) Musk



(b) Sonar



(c) DNA

Figure 2.5: Effective sample size for a fixed CPU time for sampling-based algorithms: posterior expectations (left), and posterior variances (right) for datasets (from top to bottom): Musk, Sonar, and ADN

pling, despite being a specialised algorithm, does not outperform significantly more generic algorithms. Recall that the probit/Gaussian scenario is very favourable to Gibbs sampling; in other scenarios (results not shown), Gibbs is strongly dominated by other algorithms.

More surprisingly, RWHM still performs well despite the high dimension. In addition, RHHM seems more robust than SMC to an imperfect calibration; see the DNA example, where the error of the EP approximation is greater.

On the other hand, SMC is more amenable to parallelisation, hence on a parallel architecture, SMC would be likely to outperform the other approaches.

2.6 Variable selection

We discuss in this section the implications of our findings on variable selection. The standard way to formalise variable selection is to introduce as a parameter the binary vector $\boldsymbol{\gamma} \in \{0,1\}^p$, and to define the likelihood

$$p(\mathcal{D}|\boldsymbol{\beta},\boldsymbol{\gamma}) = \prod_{i=1}^{n_{\mathcal{D}}} F(y_i \boldsymbol{\beta}_{\boldsymbol{\gamma}}^T \boldsymbol{x}_{\boldsymbol{\gamma},i})$$

where $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ (resp. $\boldsymbol{x}_{\boldsymbol{\gamma},i}$) is the vector of length $|\boldsymbol{\gamma}|$ that one obtains by excluding from $\boldsymbol{\beta}$ (resp. $\boldsymbol{x}_i$) the components j such that $\gamma_j = 0$. Several priors may be considered for this problem [Chipman et al., 2001], but for simplicity, we will take $p(\boldsymbol{\beta},\boldsymbol{\gamma}) = p(\boldsymbol{\beta})p(\boldsymbol{\gamma})$ where $p(\boldsymbol{\beta})$ is either the Cauchy prior or the Gaussian prior discussed in Section 2.2.1, and $p(\boldsymbol{\gamma})$ is the uniform distribution with respect to the set $\{0,1\}^p$, $p(\boldsymbol{\gamma}) = 2^{-p}$.

Computationally, variable selection is more challenging than parameter estimation, because the posterior $p(\boldsymbol{\beta},\boldsymbol{\gamma}|\mathcal{D})$ is a mixture of discrete and continuous components. If p is small, one may simply perform a complete enumeration: for all the 2^p possible values of $\boldsymbol{\gamma}$, approximate $p(\mathcal{D}|\boldsymbol{\gamma})$ using e.g. importance sampling. If p is large, one may adapt the approach of Schäfer and Chopin [2011], as described in the next sections.

2.6.1 SMC algorithm of Schäfer and Chopin [2011]

In linear regression, $y_i = \boldsymbol{\beta}_{\boldsymbol{\gamma}}^T \boldsymbol{x}_{\boldsymbol{\gamma},i} + \varepsilon_i$, $\varepsilon_i \sim \mathrm{N}_1(0,\sigma^2)$, the marginal likelihood $p(\mathcal{D}|\boldsymbol{\gamma})$ is available in close form (for a certain class of priors). Schäfer and Chopin [2011] use this property to construct a tempering SMC sampler, which transitions from the prior $p(\boldsymbol{\gamma})$ to the posterior $p(\boldsymbol{\gamma}|\mathcal{D})$, through the tempering sequence $\pi_t(\boldsymbol{\gamma}) \propto p(\boldsymbol{\gamma})p(\mathcal{D}|\boldsymbol{\gamma})^{\delta_t}$, with δ_t growing from 0 to 1. This algorithm has the same structure as Algorithm 9 (with the obvious replacements of the $\boldsymbol{\beta}$'s by $\boldsymbol{\gamma}$'s and so

on.) The only difference is the MCMC step used to diversify the particles after resampling. Instead of a random walk step (which would be ill-defined on a discrete space), Schäfer and Chopin [2011] use a Metropolis step based on an independent proposal, constructed from a sequence of nested logistic regressions: proposal for first component γ_1 is Bernoulli, proposal for second component γ_2, conditional on γ_1, corresponds to a logistic regression with γ_1 and an intercept as covariates, and so on. The parameters of these p successive regressions are simply estimated from the current particle system. Schäfer and Chopin [2011] show that their algorithm significantly outperform several MCMC samplers on datasets with more than 100 covariates.

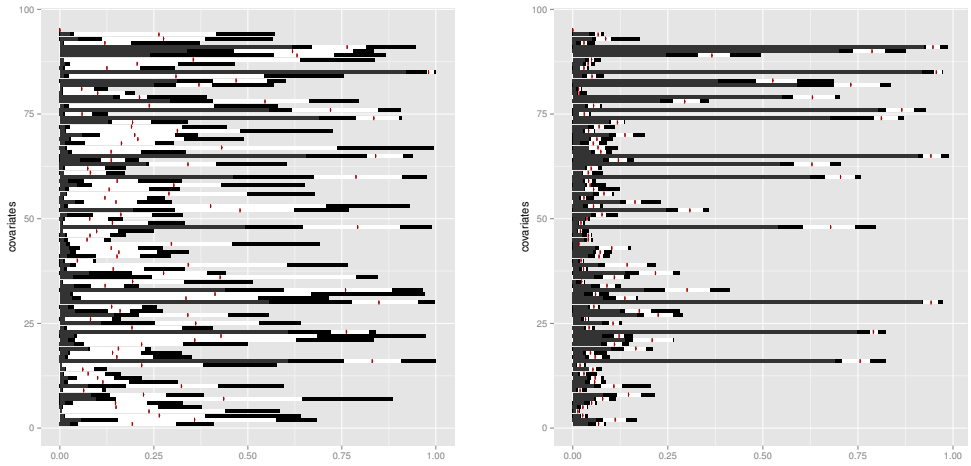
2.6.2 Adaptation to binary regression

For binary regression models, $p(\mathcal{D}|\boldsymbol{\gamma})$ is intractable, so the approach of Schäfer and Chopin [2011] cannot be applied directly. On the other hand, we have seen that (a) both Laplace and EP may provide a fast approximation of the evidence $p(\mathcal{D}|\boldsymbol{\gamma})$; and (b) both importance sampling and the tempering SMC algorithm may provide an *unbiased* estimator of $p(\mathcal{D}|\boldsymbol{\gamma})$.

Based on these remarks, Schäfer [2012] in his PhD thesis considered the following extension of the SMC algorithm of Schäfer and Chopin [2011]: in the sequence $\pi_t(\boldsymbol{\gamma}) \propto p(\boldsymbol{\gamma})p(\mathcal{D}|\boldsymbol{\gamma})^{\delta_t}$, the intractable quantity $p(\mathcal{D}|\boldsymbol{\gamma})$ is simply replaced by an unbiased estimator (obtained with importance sampling and the Gaussian proposal corresponding to Laplace). The corresponding algorithm remains valid, thanks to pseudo-marginal arguments [see e.g. Andrieu and Roberts, 2009]. Specifically, one may re-interpret the resulting algorithm as a SMC algorithm for a sequence of distribution of an extended space, such that marginal in $\boldsymbol{\gamma}$ is exactly the posterior $p(\mathcal{D}|\boldsymbol{\gamma})$ at time $t = T$. In fact, it may be seen as a particular variant of the SMC^2 algorithm of Chopin et al. [2013a].

2.6.3 Numerical illustration

We now compare the proposed SMC approach with the Gibbs sampler of Holmes and Held [2006] for sampling from $p(\boldsymbol{\beta}, \boldsymbol{\gamma}|\mathcal{D})$, on the Musk dataset. Both algorithms were given the same CPU budget (15 minutes), and were run 50 times; see Figure 2.6. Clearly, the SMC sampler provides more reliable estimates of the inclusion probabilities $p(\gamma_j = 1|\mathcal{D})$ on such a big dataset. See also the PhD dissertation of Schäfer [2012] for results consistent with those, on other datasets, and when comparing to the adaptive reversible jump sampler of Lamnisos et al. [2013].



(a) Gibbs (b) SMC

Figure 2.6: Variation of estimated inclusion probabilities $p(\gamma_j = 1|\mathcal{D})$ over 50 runs for the p covariates of Musk dataset: median (red line), 80% confidence interval (white box); the black-box extends until the maximum value.

2.6.4 Spike and slab

We also note in passing that a different approach to the variable selection problem is to assign a spike and slab prior to $\boldsymbol{\beta}$ [George and McCulloch, 1993a]:

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{p} \left\{ \lambda \mathrm{N}_1(\beta_j; 0, v_0^2) + (1-\lambda)\mathrm{N}_1(\beta_j; 0, v_1^2) \right\}, \quad v_0^2 \ll v_1^2$$

where $\lambda \in (0,1)$, v_0^2 and v_1^2 are fixed hyper-parameters. This prior generates a continuous posterior (without point masses at $\beta_j = 0$), which is easier to sample from than the discrete-continuous mixture obtained in the standard formulation of Bayesian variable selection. It would be interesting to see to which extent our discussion and findings extend to this particular type of posteriors; see for Chapter 4 for how to deal with such priors in EP.

2.7 Conclusion and extensions

2.7.1 Our main messages to users

Our first and perhaps most important message to end users is that Bayesian computation (for binary regression) is now sufficiently fast for routine use: if the right approach is used, results may be obtained near instantly on a standard computer, at least on simple datasets.

Concretely, as far as binary regression is concerned, our main recommendation is to always use EP. It is very fast, and its approximation error is negligible in most cases (for such models). EP requires some expertise to implement, but the second author will release shortly a R package that computes the EP approximation for any logit or probit model. The only drawback of EP is the current lack of theoretical support. We learnt however while finishing this manuscript that Simon Barthelmé and Guillaume Dehaene (personal communication) established that the error rate of EP is $\mathcal{O}(n_{\mathcal{D}}^{-2})$ in certain models (where $n_{\mathcal{D}}$ is the sample size). This seems to explain why EP often performs so well.

In case one wishes to assess the EP error, by running in a second step some exact algorithm, we would recommend to use the SMC approach outlined in Section 2.4.4 (i.e. with initial particles simulated from the EP approximation). Often, this SMC sampler will reduce to a single importance sampling step, and will perform extremely well. Even when it does not, it should provide decent performance, especially if run on (and implemented for) a parallel architecture. Alternatively, on a single-core machine, random walk Metropolis is particularly simple to implement, and performs surprisingly well on high-dimensional data (when properly calibrated using EP).

2.7.2 Our main message to Bayesian computation experts

Our main message to Bayesian computation scientists was already in the title of this chapter: leave Pima Indians alone, and more generally, let's all refrain from now on from using datasets and models that are too simple to serve as a reasonable benchmark.

To elaborate, let's distinguish between specialised algorithms and generic algorithms.

For algorithms specialised to a given model and a given prior (i.e. Gibbs samplers), the choice of a "benchmark" reduces to the choice of a dataset. It seems unfortunate that such algorithms are often showcased on small datasets (20 covariates or less), for which simpler, more generic methods perform much better. As a matter of fact, we saw in our simulations that even for bigger datasets Gibbs sampling does not seem to offer better performance than generic methods.

For generic algorithms (Metropolis, HMC, and so on), the choice of a benchmark amounts to the choice of a target distribution. A common practice in papers proposing some novel algorithm for Bayesian computation is to compare that algorithm with a Gibbs sampler on a binary regression posterior for a small dataset. Again, we see from our numerical study that this benchmark is of of limited interest, and may not be more informative than a Gaussian target of the same dimension. If one wishes to stick with binary regression, then datasets with more than 100 covariates should be used, and numerical comparisons should include at least a properly calibrated random walk Metropolis sampler.

2.7.3 Big data and the p^3 frontier

Several recent papers [Bardenet et al., 2015; Scott et al., 2013; Wang and Dunson, 2013] have approached the 'big data' problem in Bayesian computation by focussing on the big $n_{\mathcal{D}}$ (many observations) scenario. In binary regression, and possibly in similar models, the big p problem (many covariates) seems more critical, as the complexity of most the algorithms we have discussed is $\mathcal{O}(n_{\mathcal{D}}p^3)$. Indeed, we do not believe that any of the methods discussed in this chapter is practical for $p \gg 1000$. The large p problem may be therefore the current frontier of Bayesian computation for binary regression.

Perhaps one way to address the large p problem is to make stronger approximations; for instance by using EP with an approximation family of sparse Gaussians. Alternatively, one may use a variable selection prior that forbids that the number of active covariates is larger than a certain threshold.

2.7.4 Generalising to other models

We suspect some of our findings may apply more generally to other models (such as certain generalised linear models), but, of course, further study is required to assess this statement.

On the other hand, there are two aspects of our study which we recommend to consider more generally when studying other models: parallelisation, and taking into account the availability of fast approximations. The former has already been discussed. Regarding the latter, binary regression models are certainly not the only models such that some fast approximations may be obtained, whether through Laplace, INLA, Variational Bayes, or EP. And using this approximation to calibrate sampling-based algorithms (Hastings-Metropolis, HMC, SMC, and so on) will often have a dramatic impact on the relative performance of these algorithms. Alternatively, one may also discover in certain cases that these approximations are sufficiently accurate to be used directly.

3

Computation of Gaussian orthant probabilities in high dimension

Status: To appear in *Statistics and Computing.*

3.1 Introduction

There are many applications where computing an orthant probability in high dimension with respect to a Gaussian or Student distribution is an issue of interest. For instance it is common in statistics to compute the likelihood of models, where we observe only an event with respect to multivariate Gaussian random variables. In Econometrics, the multivariate probit model [Train, 2009], where we observe a decision among J alternative choices each of them corresponding to a Gaussian utility, is commonly studied. It can be written as an orthant problem. Other such models are the spatial probit [LeSage et al., 2011] and Thurstonian models [Yao and Bockenholt, 1999]. Other applications than direct modelization can be found, such as multiple comparison tests [Hochberg and Tamhane, 1987], where the integration is done with respect to a Student (see Bretz et al. [2001] for an example). Orthant probabilities are also of interest in other fields than statistics, i.e. stochastic programming [Prekopa, 1970], structural system reliability [Pandey, 1998], engineering, finance, etc.

The problem at hand is the computation of the integral,

$$\int_{[\mathbf{a},\mathbf{b}]} (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y-m)^t \Sigma^{-1}(y-m)\right) dy. \tag{3.1}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. The Student case will be written as a mixture of the above integral with an inverse Chi-square (see Section 3.5.1).

Many algorithms have been proposed to compute (3.1); for a review see Genz and Bretz [2009]. They can be divided into two groups. The first are numerical algorithms to deal with small dimensional integrals. In dimension 3 there exist algorithms [Genz and Bretz, 2009] where after sphericization, such that the Gaussian has an identity covariance matrix, one applies recursively numerical computations of the error function. For higher dimensions than three Minwa et al. [2003] propose to express orthant probabilities as differences of orthoscheme probabilities, where an orthoscheme is (3.1) with correlation matrix $\Omega = (\omega_{ij})$ satisfying $\omega_{ij} = 0 \quad \forall i, j \quad |i - j| > 1$. This can be easily computed by recursion. However the decomposition in orthoscheme probabilities has factorial complexity. The second group of algorithms is Monte Carlo based and may be used for dimensions higher than 10. In particular GHK due to Geweke [1991], Keane [1993] and Hajivassiliou et al. [1996] and conjointly to Genz [1992], has been widely adopted for the applications described above.

In this chapter we show that in the case of Markovian covariances (i.e. covariances that can be written as those of Markovian processes), the GHK algorithm estimates the normalizing constant of a state space model (SSM), using sequential importance sampling (SIS) with optimal proposal. We show in addition for a first order autoregressive process (henceforth AR(1)) that the normalized variance diverges exponentially fast.

To avoid this behavior we propose to use a particle filter. We extend this methodology to the non Markovian case by using Sequential Monte Carlo (SMC). SMC allows additional gain in efficiency by considering different MCMC moves and proposals. In addition the algorithm is adaptive and simplifies automatically to the GHK if the integral is simple enough. In our numerical experiments we find a substantial improvement.

We start by reviewing the existing GHK algorithm (Section 3.2), we then discuss the algorithm's behavior for Markovian covariance matrices and propose an extension to higher dimensions (Section 3.3). In Section 3.4 we extend this proposal to arbitrary covariance matrices. We propose some extensions for the simulation of truncated distributions and for other distributions (3.5). Finally we present some numerical results and conclude (Sections 3.6 and 3.7).

Notations For any vector $x \in \mathbb{R}^p$ for $i \leq p$ we write $x_{<i} \in \mathbb{R}^i$ for the vector of the $i-1$ first components, and we take $a : b = \{a, \cdots, b\}$. We let $x_{<1} = \emptyset$, and also write $x_{i:j}$ for the vector $(x_i, x_{i+1}, \cdots, x_j)$; Φ, φ are respectively the $\mathcal{N}(0,1)$ Gaussian cdf and pdf, we write $\varphi(x|A)$ for the pdf, $\frac{\varphi(x)}{\Phi(A)}\mathbb{1}_A(x)$, of a Gaussian truncated to the set $A \subset \mathbb{R}$ evaluated in x. We will also abuse notation and use $\Phi(A)$ to denote the probability of a set when $A \subset \mathbb{R}$. For instance $\Phi([a,b]) = \Phi(b) - \Phi(a)$.

3.2 Geweke-Hajivassiliou-Keane (GHK) simulator

From now on to simplify notations, and without loss of generality, we limit ourselves to the study of the following multidimensional integral:

$$F(\mathbf{a},\mathbf{b},\Sigma) = \int_{[\mathbf{a},\mathbf{b}]} (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} y^t \Sigma^{-1} y\right) dy \tag{3.2}$$

with $\mathbf{a},\mathbf{b} \in \mathbb{R}^d$. Note that the extension to integrals where some components of the vectors $\mathbf{a},\mathbf{b}$ are respectively $-\infty$ and ∞ is direct.

Let Γ be the Cholesky decomposition of Σ, i.e. $\Sigma = \Gamma\Gamma^t$ with $\Gamma = (\gamma_{ij})$, $\gamma_{ii} > 0$ and $\gamma_{ij} = 0$ if $j > i$. We can write the previous equation after the change of variable $\eta = \Gamma^{-1} y$ for which $d\eta = |\Gamma|^{-1} dy$:

$$F(\mathbf{a},\mathbf{b},\Sigma) = \int_{\mathbf{b}\geq\Gamma\eta\geq\mathbf{a}} (2\pi)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}\eta^t\eta\right) d\eta,$$

the i-th truncation being such that $\frac{1}{\gamma_{ii}}\left(a_i - \sum_{j=1}^{i-1}\gamma_{ij}\eta_j\right) \leq \eta_i \leq \frac{1}{\gamma_{ii}}\left(b_i - \sum_{j=1}^{i-1}\gamma_{ij}\eta_j\right)$, from the positivity of the (γ_{ii}). Thus we can write:

$$F(\mathbf{a},\mathbf{b},\Sigma) = \int \prod_{i=1}^{d} \varphi(\eta_i)\mathbf{1}_{\{B_i(\eta_{<i})\}}(\eta_i) d\eta_{1:d} = \int \prod_{i=1}^{d} \Phi\left(B_i(\eta_{<i})\right) \varphi(\eta_i | B_i\left(\eta_{<i}\right)) \, d\eta_{1:d},$$

where the set $B_i(\eta_{<i}) = \{\eta_i : \frac{1}{\gamma_{ii}}\left(a_i - \sum_{j=1}^{i-1}\gamma_{ij}\eta_j\right) \leq \eta_i \leq \frac{1}{\gamma_{ii}}\left(a_i - \sum_{j=1}^{i-1}\gamma_{ij}\eta_j\right)\}$ is an interval.

The GHK algorithm is an importance sampling algorithm based on this structure. It proposes particles distributed under $\prod_{i=1}^{d}\varphi(\eta_i|B_i\left(\eta_{<i}\right))$ and evaluates the average of the weights $w^n = \prod_{i=1}^{d}\Phi\left(B_i(\eta^n_{<i})\right)$. The algorithm is described in pseudo-code in Alg. 10.

Algorithm 10 GHK simulator

for $m \in 1:M$ **do**
  Sample: $\eta^m_{1:d} \sim \prod_{i=1}^{d}\varphi(\eta_i|B_i(\eta_{<i}))$
  Weights*: $w^m = \prod_{i=1}^{d}\Phi\left(B_i(\eta^m_{<i})\right)$
end for
return $\frac{1}{M}\sum_{i=1}^{M} w^i$

*Recall that $\Phi(B_i(\eta^m_{<i})$ can be computed as a difference of two one dimensional cdf for the truncation defined above.

Algorithm 10 outputs an unbiased estimator of integral (3.1).

To generate truncated Gaussian variables the usual approach in the GHK simulator is to use the inverse cdf method. We follow this approach in the rest of the chapter except where stated otherwise. When the numerical stability of the inverse cdf is an issue we will use the algorithm proposed in Chopin [2011a].

In the next section we will study with more care the case where the covariance matrix of the underlying Gaussian vector has a Markovian structure.

3.3 The Markovian case

When the covariance matrix is Markovian, that is a matrix for which the inverse is tri-diagonal, the simulation step of Alg. 10 is the simulation of a Markov process $(x_{1:t})$. At time t the weights depend on x_{t-1} only. Let us take a lag 1 autoregressive process (AR(1)) for the purpose of exposition, and study the probability of it being in some hyperrectangle $[\mathbf{a}, \mathbf{b}] = [a_1, b_1] \times \cdots \times [a_T, b_T]$. The integral of interest is therefore:

$$\int \prod_{t=1}^{T} \mathbf{1}_{\{[a_t,b_t]\}}(x_t)\varphi(x_t; \varrho_t x_{t-1}, \sigma_t^2)dx_{1:T}. \tag{3.3}$$

The GHK algorithm consists in sampling from the Markov process:

$$x_t|x_{t-1} \sim \varphi\left(\varrho_t x_{t-1}, \sigma_t^2|B_t(x_t)\right).$$

The matrix Σ^{-1} is tridiagonal, the weights at time t are therefore $\Phi(\frac{b_t-\varrho_t x_{t-1}}{\sigma_t}) - \Phi(\frac{a_t-\varrho_t x_{t-1}}{\sigma_t})$. Eq. 3.3 can be seen as the likelihood of the state space model [Cappé et al., 2005]:

$$\begin{aligned} x_t|x_{t-1} &\sim \varphi(x_t; \varrho_t x_{t-1}, \sigma_t^2) \\ y_t|x_t &\sim \mathbf{1}_{\{[a_t,b_t]\}}(x_t) \end{aligned}$$

where $(y_t)_t$ is observed. The GHK can be interpreted as a sequential importance sampler (SIS) using proposal $\varphi(\varrho_t x_{t-1}, \sigma_t^2|B_t(x_t))$.

3.3.1 Toy example

Let us specify a bit more the problem to simplify notation and show some properties of a thus defined algorithm.

Consider the problem of finding the probability that an AR(1),

$$X_t = \varrho X_{t-1} + \varepsilon_t, \qquad |\varrho| < 1$$

is inside the hyper-cube $[0, b] \times \cdots \times [0, b]$, for some $b > 0$. We have set $\sigma = 1$, $a = 0$ and ϱ a constant.

The GHK algorithm consists in this case in simulating the above Markov chain constrained to $[0, b]$ and in computing under this distribution the products of the weights $\prod_{t=1}^{T} [\Phi(b - \varrho X_t) - \Phi(-\varrho X_t)]$. The simulations are therefore generated by the Markov probability kernel

$$P^b(x, dy) = \frac{\varphi(y; \varrho x, 1)}{\Phi(b - \varrho x) - \Phi(-\varrho x)} \mathbb{1}_{[0,b]}(y) dy, \quad |\varrho| < 1. \tag{3.4}$$

For this model we have the following proposition:

Proposition 3.3.1 *For the Markov model defined by* (3.4)*, the normalized square product of weights of the normalizing constant has the following behavior:*

$$\liminf_{T\to\infty} \left\{ \mathbb{E}\left[\left(\frac{\prod_{t=1}^{T} \left(\Phi\left(b - \varrho X_t\right) - \Phi\left(-\varrho X_t\right)\right)^2}{\exp\{2T\mathbb{E}_\pi \log\left(\Phi(b - \varrho X) - \Phi(-\varrho X)\right)\}} \right) \right] \right\}^{\frac{1}{\sqrt{T}}} > \exp\left\{\mathbb{V}_\pi\left[\psi(X)\right] + \tau\right\}, \tag{3.5}$$

where subscript π denotes integration with respect to the invariant distribution of $P^b(x, \mathrm{d}y)$, the other expectation is taken relatively to the Markov chain (X_t), and $\psi : x \mapsto \log\left(\Phi(b - \varrho x) - \Phi(-\varrho x)\right)$, $\tau = 2\sum_{k=1}^{\infty} cov(X_0, X_k)$.

Proof: A detailed proof is given in appendix 3.A. □

Under V-Uniform ergodicity, that follows from our proof, the denominator is the square of the limit of the product of weights and can be interpreted as a scaling factor. Thus the result above shows that this renormalized squared estimator diverges exponentially fast as the dimension of the integral increases.

Remark 3.3.1 *In the course of the proof we showed that the normalizing constant has a log-normal limiting distribution, resulting in a skewed distribution. We expect that the distribution of the estimator will have its mode away from the expected value resulting in some apparent bias. In fact one can show that the normalized third order moment will also grow exponentially.*

GHK has quadratic complexity however we can show that for at least one covariance structure the variance diverges exponentially fast. This fully justifies the use of an algorithm of higher computational complexity. In the following section we propose a natural extension to deal with this issue in the Markovian case.

3.3.2 Particle filter (PF)

PF is a common extension of SIS that corrects the weight degeneracy problem. The solution brought by particle filtering [Gordon et al., 1993] is to use a resampling

step, i.e. to kill those particles with low weights and to replicate those with high contribution. At time t one resamples the particles by sampling from the distribution $\sum_{m=1}^{M} W_t^m \delta_{x_t^m}(\mathrm{d}x)$ where W_t^m stands for the m-th renormalized weight at time t, and $\delta_x(dx')$ the Dirac measure in x. All the weights are then set to one.

We use an adaptive version of this algorithm where the resampling step is triggered only when the ESS of the weight is lower than some threshold, where the ESS is defined as

$$\frac{\left(\sum_{i=1}^{N} w_i\right)^2}{\sum_i w_i^2} \in [1, M]\,,$$

and indicates the number of draws from the independent distribution to obtain the same variance. Note that it is closely related to the inverse of equation (3.5), hence we expect that without resampling it goes to zero with exponential speed.

We define the state space model:

$$x_t|x_{t-1} \sim g_t(x_t|x_{t-1}),$$
$$y_t|x_t \sim f_t(y_t|x_t)$$

One can use a PF to compute the likelihood of such model,

$$L(y_{1:T}) = \int \prod_{t=1}^{T} g_t(x_t|x_{t-1}) f_t(y_t|x_t) g_0(x_0) dx_{0:T}$$

A PF with proposal distribution $q_t(x_t, x_{t-1})$ is described in Alg. 23. Our application corresponds to the special case where:

$$g_t(x_t|x_{t-1}) = \varphi(x_t), \qquad f_t(y_t|x_t, x_{t-1}) = \mathbf{1}_{\{B_t(x_{<t})\}}(x_t), \qquad q_t(x_t|x_{t-1}) = \varphi(x_t|B_t(x_{<t})),$$

where the set $B_t(x_{<t})$ depends on x_{t-1} only.

The proposal thus defined corresponds to the optimal one [Doucet et al., 2000], that is the distribution proportional to $f_t(y_t|x_t)g_t(x_t|x_{t-1})$ in our case proportional to $\varphi(x_t)\mathbf{1}_{\{B_t(x_{<t})\}}(x_t)$ hence the truncated Gaussian. The weights are given by the normalizing constant $\int f_t(y_t|x_t)g_t(x_t|x_{t-1})\mathrm{d}x_t$, in our case $\Phi(B_t(x_{<t}))$.

To resample we propose to use systematic resampling [Carpenter et al., 1999] (for other approaches see Douc et al. [2005]). Systematic resampling is described in Algorithm 13 (Appendix 3.B).

The particle filter thus defined outputs an unbiased estimator of the likelihood Del Moral [1996a], and thus the orthant probability in our case.

Note that the output of Algorithm 23 is of the form of a product of terms smaller than one, in our case those terms can be very small and lead to numerical issues. One way of dealing with this issue is to rewrite all the algorithm in log scale.

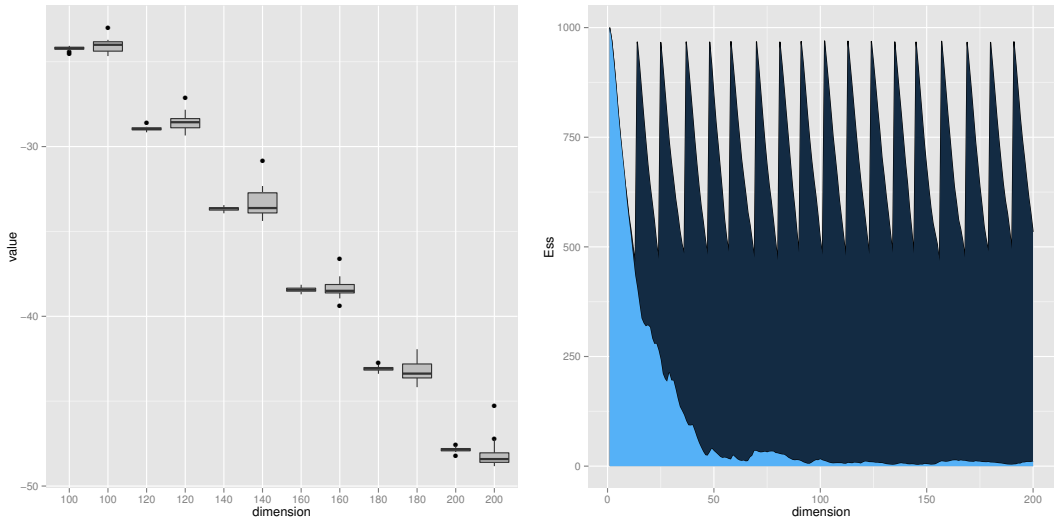
Algorithm 11 Particle Filter

Input: M the number of particles
Sample: Sample $x_0^i \sim g_0(.)$
for $t = 1 : T-1$ **do**
  if $ESS < \eta^\star$ **then**
    $Z \leftarrow Z \times \{\frac{1}{M}\sum_{i=1}^M w_t^i\}$
    Resample $a_t^j \sim \sum_i \frac{w_t^i}{\sum_j w_t^j}\delta_i$ using algorithm 13, set $w_t^j \leftarrow 1$
  else
    $a_t^{1:M} = 1 : M$
  end if
  Sample $x_{t+1}^i \sim q_{t+1}(.|x_t^{a_t^i})$
  Set $w_{t+1}^i \leftarrow w_t^i \frac{f_{t+1}(y_{t+1}|x_{t+1}^i) g_{t+1}(x_{t+1}^i|x_t^{a_t^i})}{q_{t+1}(x_{t+1}^i|x_t^{a_t^i})}$
end for
return $Z \times \frac{1}{M}\sum_{i=1}^M w_T^i$

Remark 3.3.2 *As we are here in the special case of being able to sample from the optimal distribution (as shown in Section 3.3) one could resort to the auxiliary particle filter (APF, Pitt and Shephard [1999]). In fact in this special case the algorithm amounts to exchanging the resampling step and the move step of the particle filter. We tested this approach on some Markov processes and observed no improvements in term of variance on repeated draws.*

Example 3.3.1 *We can show that the previous process (Section 3.3.1) benefits from resampling when the ESS goes beneath a given level.*

Figure (3.1) shows that the GHK algorithm's variance increases more quickly as compared to the PF (that seem to have some stable variance on the considered dimension). In addition the distribution of the GHK estimator seem to be skewed towards smaller values as T increases. This results in some bias on the last boxplot. As described in remark 3.3.1 this behavior is due to the log-Normal limiting distribution of the output of the algorithm. The skewness coefficient increases exponentially with T.



(a) variance (b) ESS

Figure 3.1: Estimates of Orthant probabilities by Particle Filter

Estimation of the log probability that an AR(1) process (defined previously) with $\varrho = 0.7$ has all its component in $[0, 15]$. GHK sampler (grey) and PF (white) on various dimension from 100 to 200. On the right panel the two ESS for dimension 200. On both cases M is set to 1000.

Thurstonian Model

Thurstonian models arise in Psychology and Economics [Yao and Bockenholt, 1999] to describe the ranking of p alternatives by n individuals (referred to as judges).

Suppose that we observe the rank $r_i = (k_{1i}, \cdots, k_{p,i})$ of some p independent Gaussian random variables,

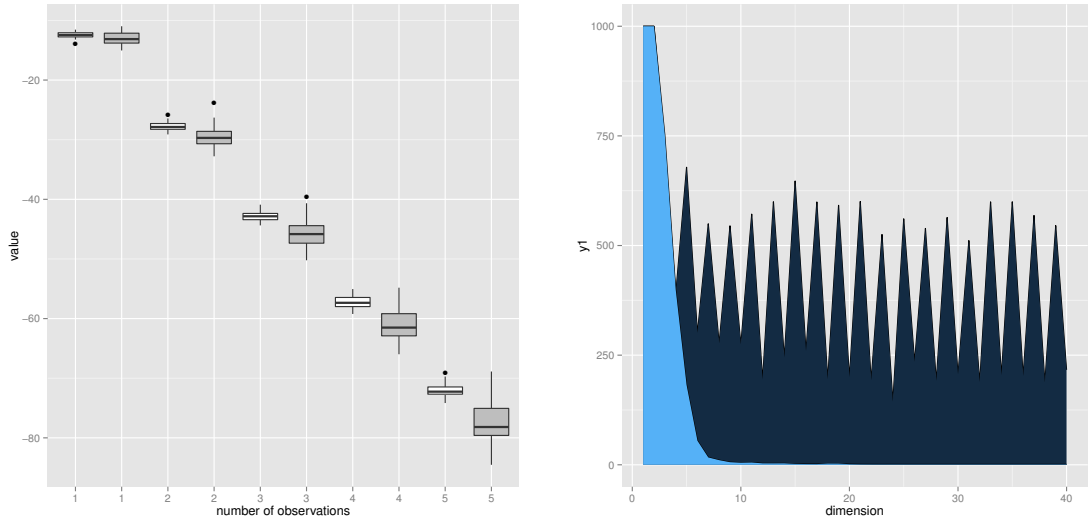
$$x_{i,j} = \beta_j + \sigma\varepsilon_{i,j},$$

where $\varepsilon_{i,j} \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$. The likelihood of one observation is an orthant probability:

$$\mathbb{P}_\theta\{X_p > \cdots > X_1\} = \int \prod_{i=1}^{p} \mathbf{1}_{\{x_i > x_{i-1}\}} \varphi(x_i|\beta_j, \sigma^2) dx_{1:p} \tag{3.6}$$

with the convention that $X_0 = -\infty$.

This model is similar to the previous one but with $\rho = 1$.



(a) Estimates (b) ESS

Figure 3.2: Estimates of the likelihood of a Thurstonian model by Particle Filter and by GHK

Estimation of the likelihood of a Thurstoninan model with $p = 10$ and the number of observations is ranging from 1 to 5 the PF (white) and the GHK (grey). The threshold ESS is set to $0.5M$ and the number of particles is set to $M = 1000$. The right panel shows the ESS of both algorithms for $T = 1$ and $p = 40$.

We find that the likelihood is estimated with smaller variance. In addition, because of the heavy tail distribution of the GHK simulator's output we observe a bias (see Figure 3.2). Again one can explain the strong observed bias by remark 3.3.1 and the fact that we do not replicate enough the experiment to observe the tail of the distribution. In addition as suggested above the ESS of the GHK seems to decrease exponentially fast to zero.

From this observation we could apply this algorithm to perform inference by using Particle MCMC [Andrieu et al., 2010a], where this estimation of the likelihood can be plugged in a Random walk Metropolis Hastings and still target the appropriate distribution.

3.4 Non Markovian case

For more general covariances we propose to use Sequential Monte Carlo (SMC) [Del Moral et al., 2006b]. As previously we will base the algorithm on the proposal

of GHK, increasing the dimension of the problem at each time step. However we now have an additional degree of freedom: the order in which we incorporate the variables. In the following section we study an approach to ordering the variables.

3.4.1 Variable ordering

We follow Gibson et al. [1994] in ordering the variables from the most difficult to the simplest, where difficult constraints are considered to be the one that impact the most the probability.

However we cannot evaluate exactly the probabilities as it is our final goal. Instead Gibson et al. [1994] propose to replace the simulations by the expected value of the truncated Gaussian.

The algorithm starts by choosing the first index i_1, and defining η_1 as follows:

$$i_1 = \arg\min_{1\leq k\leq T} \Phi\left(\left[\frac{a_k}{\gamma_{kk}}, \frac{b_k}{\gamma_{kk}}\right]\right), \qquad \eta_1 = \frac{1}{\Phi\left(\left[\frac{a_{i_1}}{\gamma_{i_1 i_1}}, \frac{b_{i_1}}{\gamma_{i_1 i_1}}\right]\right)} \int_{\left[\frac{a_{i_1}}{\gamma_{i_1 i_1}}, \frac{b_{i_1}}{\gamma_{i_1 i_1}}\right]} \eta\varphi(\eta)d\eta$$

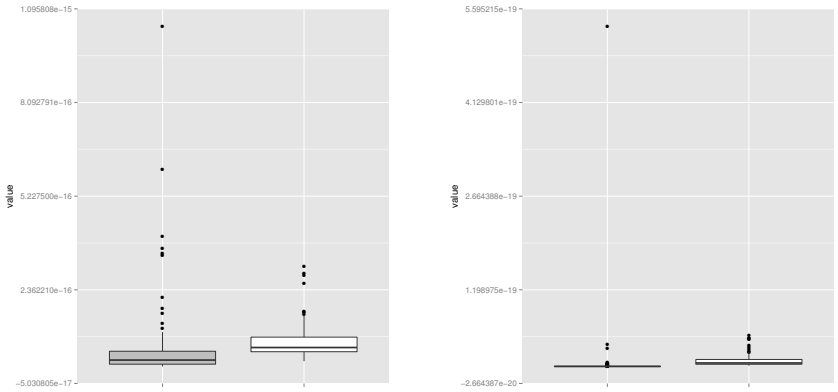
i.e. the smallest possible probability that the Gaussian will be in $[a_k, b_k]$. This enables an approximation of the next probability as a function of i_2.

$$i_2 = \arg\min_{2\leq k\leq T} \Phi\left(\left[\frac{1}{\tilde{\gamma}_{kk}}(a_k - \tilde{\gamma}_{1,k}\eta_1), \frac{1}{\tilde{\gamma}_{kk}}(b_k - \tilde{\gamma}_{1,k}\eta_1)\right]\right).$$

where $(\tilde{\gamma}_{ij}) = \tilde{\Gamma}$ is the Cholesky decomposition of the matrix after substituting the first and the i_1th variable.

We end up with the desired vector $(i_1, \cdots, i_T)$ that gives us the order in which to choose the covariances and truncation points. The algorithm is summed up by Alg. 14 in appendix 3.C. The algorithm has quadratic time complexity, however its cost is negligible as compared to the subsequent Monte Carlo algorithm.

We show the use of the reordering in moderate dimensions (50 and 60) on the GHK simulator. This is already a great improvement especially as the dimension increases. Figure (3.3), shows boxplots of 50 repetitions of the GHK for both ordered (white) and non-ordered (grey) inputs.



(a) Dimension 50 (b) Dimension 60

Figure 3.3: Estimates of Orthant probabilities with (white) and without (grey) variable ordering

Covariance matrices generated from random samples with heavy tails (see Section 3.6). In both case we use a GHK simulator with variable ordering (white), without (grey). The various dimension are simulated with the same algorithm and same seed, such that the small ones are subsets of the others. When the variables are not ordered we observe some outliers, and this phenomenon is reduced with Gibson et al. [1994]'s algorithm.

From dimension 50 and upwards we start observing some skewed distributions for the GHK estimator as noted in remark 3.3.1. The phenomenon seems to be reduced by ordering.

This effect is relatively dangerous as some draws depart a lot from the mean value. The ordering will be used on all examples from now on to reduce the variance. In Section 3.6 we have empirical evidence that using an appropriate move step deals with this tail effect, in our examples.

We have shown that in the particular case of Markov processes we can strikingly benefit from the use of resampling. In the next section we attempt to generalize our finding to a broader range of problems.

3.4.2 A sequential Monte Carlo (SMC) algorithm

The algorithm discussed in Section 3.3 can be generalized to non Markovian Gaussian vectors by applying the SMC methodology. Define the following sequence of distribution:

$$\pi_t(\eta_{1:t}) = \frac{\gamma_n(\eta_{1:t})}{Z_t}, \qquad \gamma_t(\eta_{1:t}) = \prod_{i=1}^{t} \varphi(\eta_i)\mathbf{1}_{\{B_i(\eta_{<i})\}}, \tag{3.7}$$

indexed by t, where the unnormalized quantity $\gamma_t(\eta_{1:t})$ is our target integrand. We thus want to compute an estimator of Z_t for a given t,

$$Z_t = \int \prod_{i=1}^{t} \varphi(\eta_i) \mathbf{1}_{\{B_i(\eta_{<i})\}} \mathrm{d}\eta_{1:t}.$$

SMC samplers are a class of algorithms that generalize particle filters to non dynamic problems (Neal [2001a],Chopin [2002a], Del Moral et al. [2006b]). Their aim is to sample from a sequence of measures $(\pi_t)_t$ where π_0 is easy to sample from and π_T is our target. The algorithm works by moving from one target to the other by importance sampling, and avoid degeneracy of the weights by resampling if the ESS falls bellow a threshold. In the case of the GHK the sequence of distribution consist of adding a dimension at each step. To ensure particle diversity after resampling the particles are moved according to a MCMC kernel targeting the current distribution. This is the most computationally expensive step. Different alternatives are described in the next section.

The main steps are described in Algorithm (12) bellow:

Algorithm 12 SMC for orthant probabilities

Input Γ,a, M, $\alpha^\star$, Set $Z \leftarrow 1$
Each computation involving m is done $\forall m \in 1:M$
Init $\eta_1^m \sim \varphi(.)\mathbf{1}_{B_k(\eta_{<1})}$, and $w_1^m = 1$
for $t \in 1:T-1$ **do**
  At time t the weighted system is distributed as $(w_t^m, \eta_{1:t}^m) \sim \prod_{k=1}^{t} \varphi(\eta_k)\mathbf{1}_{B_k(\eta_{<k})} \propto \pi_t(\eta_{1:t})$.
  if $ESS(w_t^{1:M}) < \alpha^\star$ **then**
    $Z \leftarrow Z \times \{\frac{1}{M}\sum_{i=1}^{M} w_t^i\}$.
    Resample: $\eta_t'^m \sim \sum_{j=1}^{M} w_t^j \delta_{\eta_t^j}$, $w_t^m \leftarrow 1$.
    Move: $\eta_t^m \sim K_t(\eta_t'^m, d\eta_t^m)$ where K_t leaves $\pi_t(\eta_{1:t})$ invariant.
  end if
  $\eta_{t+1}^m \sim \varphi(.|B_{t+1}(\eta_{<t+1}^m))$, $w_{t+1}^m \leftarrow w_t^m \times \Phi(B_{t+1}(\eta_{<t+1}))$.
end for
return $Z \times \{\frac{1}{M}\sum_{i=1}^{M} w_T^i\}$

An interesting feature of Algorithm 12 is that if the integral is simple enough the ESS will never fall under the threshold and the above algorithm breaks down to a GHK simulator. This allows the algorithm to adapt to simple cases at a minimal effort, that of computing the ESS.

Note also that the estimator is still unbiased (Del Moral [1996a]) and can therefore be used in more complex schemes such as PMCMC (Andrieu et al. [2010a]) or SMC^2 (Chopin et al. [2013b]).

3.4.3 Move steps

The moves step will have an important impact on the non-degeneracy of the particle system. We want to construct a Markov chain that moves the particles as far away from their initial position as possible. In addition this step will be the bulk of the added time complexity compared to GHK, so we want to make it as efficient as possible.

Gibbs sampler

The structure of our target (3.7), where the dependence of the Gaussian components lies within the truncation, does not allows a direct application of the Gibbs sampler of Robert [1995], without a change of variable. In this section, to simplify notations we consider the special case of $\mathbf{b} = \infty$. We write the conditional distribution at time t as proportional to $\prod_{i=1}^{t} \varphi(\eta_i) \mathbb{1}_{(\Gamma_{<t,<t}\eta_{<t}) > b_{<t}}$, where $\Gamma_{<t,<t}$ is the matrix built with the first $t-1$ lines and columns of Γ. The conditional is given by:

$$\eta_i | \eta_{-i} \sim \varphi \left(. \Big| \bigcap_{j \geq i}^{t} \left\{ \operatorname{sign}(\gamma_{ji}) \eta_i \geq \frac{1}{|\gamma_{ji}|} \left(a_j - \sum_{k \neq i} \eta_k \gamma_{jk} \right) \right\} \right).$$

We therefore have to compute those sets for each component up to t and simulate according to a truncated Gaussian. Computing the set can lead to one or two sided truncations depending on the sign of the γ_{ij}.

The main drawback about having to compute this step each time we resample is its complexity. This operation has time complexity $\mathcal{O}(d^3)$ per time step. This is easily seen as the set in the above equation is just the result of some matrix inversion for a lower triangular system of dimension d. This leads to an SMC algorithm that seem to have a prohibitive complexity of $\mathcal{O}(d^4)$, where the GHK simulator had an $\mathcal{O}(d^2)$ complexity. However we have shown that GHK's variance diverges exponentially quickly on some examples suggesting that this complexity might be acceptable. In fact examples in high dimension show that even at constant computational cost the algorithm is able to out-perform GHK (see Section 3.6).

Hamiltonian Monte Carlo

An alternative to Gibbs sampler is to use Hamiltoninan Monte Carlo (HMC) (see [Neal, 2010a] for a survey), and the idea of Pakman and Paninski [2012] for truncated Gaussians.

HMC is based on interpreting the variables of interest as the position of a particle with potential the opposite of the log target and by simulating the momentum as a Gaussian with given mass matrix. The proposal of the Metropolis-Hastings is then

constructed by applying the equations of motion up to a time horizon T_{HMC} to the problem. This leads to an efficient algorithm that makes use of the gradient of the target to explore its support. We refer the reader to Neal [2010a] for more details on the algorithm and describe the approach proposed by Pakman and Paninski [2012] to adapt the algorithm to truncated Gaussians.

Based on the fact that the log density of a Gaussian random variable is a quadratic form, the movement equation can be dealt with explicitly. The scheme is written as an exact HMC (i.e. not resulting in numerical integration). Remains then to deal with the truncation. Pakman and Paninski [2012] show that they can be treated as "walls" for the given particle, a reflection principle can be applied for any particle hitting the constraint during the algorithm. In particular we must find the time at which occurs the first "hit". In our experiment the time horizon T_{HMC} is set to a uniform draw on $[0, \pi]$ as suggested in Neal [2010a]. The average value $\pi/2$ is advocated by Pakman and Paninski [2012].

The computation of the first hitting time dominates the cost of the algorithm. This is particularly true when the truncation are small as the number of hitting times will be high. Figure 3.D.1 in appendix 3.D shows a comparison of the SMC algorithm with the Gibbs sampler (grey) and exact HMC (white). Although this Markov chain algorithm seems to perform very well for a wide range of problems and has a neat formalism, we find that it does not outperform Gibbs sampling when used as a move. The specificity of the move step in SMC is that the particles are already distributed according to (3.7), therefore the move need not propagate each particle across all the support. In particular the strength of HMC in quickly exploring the target might be less useful in this context.

Overrelaxation

Overrelaxation for Gaussian random variables was proposed by Adler [1981] as a way of improving Gibbs sampling for a distribution with Gaussian conditionals.

For each component the proposal is $\eta_i'|\eta_{-i} \sim \mathcal{N}(\mu_i + \alpha(\eta_i - \mu_i), \sigma_i^2(1-\alpha^2))$ for $0 \leq \alpha \leq 1$, and with μ_i and σ_i^2 the expectation and variance of $\eta_i|\eta_{-i}$. The case $\alpha = 0$ is the classical Gibbs sampler, the case $\alpha = 1$ is a special case of random walk Metropolis-Hasting proposal. One can check that if $\eta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ then η_i' has the correct distribution.

Given a particle η, we propose a new one according to:

$$\eta'|\eta \sim \mathcal{N}(\alpha\eta, (1-\alpha^2)I)$$

Setting aside the constraint for a moment the invariant distribution of such kernel is an independent (0,1)-Gaussian. If we add an acceptation step such that we accept if it satisfies the constraint at time t, the Markov kernel leaves the current distribution invariant (3.7).

We find that the fact that overrelaxation is close to a Metropolis adjusted Langevin algorithm (MALA) helps to calibrate the algorithm, $\log \pi(\eta) = -\frac{1}{2}\eta^T\eta$, hence the proposal in MALA is $\mathcal{N}((1-\frac{\varepsilon}{2})\eta, \varepsilon^2)$. From Roberts and Rosenthal [1998] we have that ε should be $\mathcal{O}(d^{-\frac{1}{3}})$. To calibrate the algorithm we propose to match the two drifts. We find that $\alpha = \mathcal{O}(1-0.5d^{-\frac{1}{3}})$, the constant should then be close from a problem to an other because locally we are always in the case of independent Gaussians (locally the constraints have less impact). We find that in our case taking $\alpha = 0.004 \times (1-d^{-1/3})$ gives the expected behavior and acceptance ratio.

Repeating the move step

Dubarry and Douc [2011] have shown, for particle filters, that applying some Metropolis Hastings kernel targeting the filtering distribution on the particles leads to a close to optimal variance (the variance is the same as one coming from an *iid* sample). This convergence results happens after $\mathcal{O}(\log M)$ iteration of the Markov kernel. These results suggest repeating the move step after each resampling step until some criterion of convergence is satisfied.

We compute the sum of absolute distances that the particles have moved after each step (a similar metric was used in Schäfer and Chopin [2011] for the discrete case). We repeat the move until this scalar value stabilizes. The stabilization of the total metric should be associated with the cancellation of the dependence between the particles (leading to a close to independent system).

Block sampling

To diversify the particle system after each resampling we have relied until now on invariant kernels targeting the current distribution π_t. An alternative to this approach is given by Doucet et al. [2006], where importance sampling is done on the space of $\eta_{t-L+1:t+1}$ with a given number L of previous time steps. This limits the behavior of the particles all stemming from one path after a few iterations. We briefly describe the idea in the following.

Suppose at time $t-1$ we have a weighted set of particles such that $(w_{t-1}, \eta_{1:t-1}) \sim \pi_{t-1}(\eta_{1:t-1})$; instead of proposing a particle η'_t, propose a block of size L, $\eta'_{t-L+1:t} \sim q(.|\eta_{1:t-1})$, and discard the particles $\eta_{t-L+1:t-1}$. The distribution of the resulting system is intractable because of the marginalization. However Doucet et al. [2006] note that importance sampling is still possible on the extended set of particles $(\eta_{1:t-1}, \eta'_{t-L+1:t})$ by introducing some auxiliary distribution $\lambda_t(\eta_{t-L+1:t-1}|\eta'_{1:t})$. This leads to the correct marginal whatever λ_t and the algorithm has the following

incremental weights:

$$\frac{\pi_t(\eta_{1:t-L}, \eta'_{t-L+1:t})\lambda_t(\eta_{t-L+1:t-1}|\eta'_{t-L+1:t}, \eta_{1:t-1})}{\pi_{t-1}(\eta_{1:t-1})q(\eta'_{t-L+1:t}|\eta_{1:t-1})}.$$

The authors show that the optimal proposal and resulting weights are given by:

$$q^{opt}(\eta'_{t-L+1:t}|\eta_{1:t-1}) = \pi_t(\eta'_{t-L+1:t}|\eta_{1:t-L}),$$

$$w_t = w_{t-1}\frac{\pi_t(\eta'_{1:t})}{\pi_{t-L}(\eta_{1:t-L})}.$$

In our case the optimal proposal can then be shown to be:

$$q_t^{opt}(\eta'_{t-L+1:t}|\eta_{1:t-L}) = \frac{\prod_{i=1}^t \mathbb{1}_{B_i(\eta_{<i})} \prod_{i=t-L+1}^t \varphi(\eta_i)}{\int \prod_{i=1}^t \mathbb{1}_{B_i(\eta_{<i})} \prod_{i=t-L+1}^t \varphi(\eta_i) d\eta_{t-L+1:t}}.$$

Notice that this is the density of a truncated Gaussian distribution, yielding a weight depending on an orthant probability (denominator). In most cases this is not available and in our particular case it is the quantity of interest. We can however compute explicitly this integral for $L = 1$ and $L = 2$. The former is the usual case (block of size one). The case $L = 2$ did not bring any improvement in terms of variance in all our simulation. We concentrated on the extension to blocks of higher dimension.

In this case we have to resort to approximations of the proposal. The first idea would be to approximate it by a Gaussian using expectation propagation [Minka, 2001a]. However this approach did not perform better than the use of Gibbs sampler mentioned earlier. Another approach to approximate the distribution is to consider the Gibbs sampler on a block of size L with the GHK proposal.

Partial conclusion

We have shown that the proposed Gibbs sampler outperforms HMC. Concerning block sampling the different approaches were tested on several dimensions only to find that the best performing approach was to use partial Gibbs sampling, i.e. a Gibbs sampler on a block. In the numerical tests we provide in Section 3.6 we show only the latter.

In our simulations we propose to repeat each kernels as was explained in Section 3.4.3. We propose to test the Gibbs sampler and the overrelaxed random walk.

In addition we have studied other kernels based on the geometry of the problem; in particular, one can draw random walks on the line between the current particle and the basic solution of our constraint. Those approach did not however outperform the proposals discussed above.

3.5 Extentions

3.5.1 Student Orthant

We can easily extend our approach to the computation of orthant probabilities for other distributions, in particular for mixtures of Gaussians, that is probabilities that can be written as:

$$\int f_U(u) \int f_{H|U}(\eta|u)\mathbf{1}_{\{b>\eta>a\}}d\eta du, \tag{3.8}$$

where $f_{H|U}$ is a Gaussian. Several distributions can be created as such. For instance, the Student distribution where the variance is marginally distributed as an inverse-χ^2. Hence the distribution:

$$f_{H|U}(\eta|u)\mathbf{1}_{\{b>\eta>a\}} = \prod_{i=1}^{n} \varphi(\eta_i)\mathbf{1}_{\{B_i^u(\eta_{<i})\}},$$

where $B^u(\eta_{<i})$ is $B_i(\eta_{<i})$ where we multiply a by $\frac{u}{\nu}$ and $f_U(u) = \chi^2_\nu(u)$. They are an interesting application to those algorithms because they come at a minimal additional cost and are of use in multiple comparison [Bretz et al., 2001].

Another example is the logistic distribution where $f_U(u)$ is some transformation of a Kolmogorov-Smirnov distribution (see Holmes and Held [2006]). This could be used to perform Bayesian inference on multinomial logistic regression.

To deal with this integral we can extend the space on which the SMC is carried out at time t. Hence the move step is performed on the extended space $f_U(u)f_{H|U}(\eta|u)$. In our Student example it amounts to taking as a target distribution

$$\pi_n(\eta_{1:n}, u) \propto \prod_{i=1}^{n} \varphi(\eta_i)\mathbf{1}_{\{B_i^u(\eta_{<i})\}}\chi^2_\nu(u).$$

The normalizing constant that the SMC algorithm approximates is

$$Z_n = \int \prod_{i=1}^{n} \varphi(\eta_i)\mathbf{1}_{\{B_i^u(\eta_{<i})\}}\chi^2_\nu(u)\mathrm{d}\eta_{1:n}\mathrm{d}u.$$

At each move step we therefore move the particles using a Metropolis-Hastings algorithm targeting $p(u|\eta_{1:n})$ and perform the remaining Gibbs sampler updates conditionally on U. This additional step allows for further mixing. Benefits from this step are already found in relatively low dimension as shown in Section 3.6.

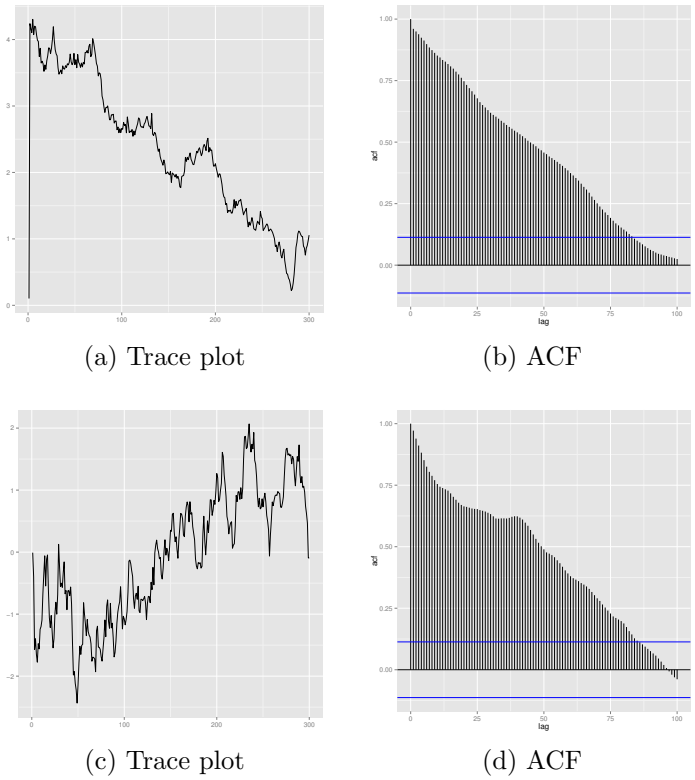
3.5.2 SMC as a truncated distribution sampler

A natural extension is to use Alg. 12 to compute other integrals with respect to truncated Gaussians. At time t the output of the algorithm is a weighted sample $(w_t^i, \eta_{1:t}^i)_{i\in[1,M]}$ approximating $\pi_t(\eta_{1:t}) \propto \prod_{i=1}^t \varphi(\eta_i)\mathbb{1}_{B(\eta_{<i})}$. Hence any integral of the form $\mathbb{E}_{\pi_t}(h(\eta))$, where expectation is taken with respect to π_t, can be approximated by $\sum_{i=1}^M \frac{w_t^i}{\sum_{j=1}^M w_t^j} h(\eta_{1:t}^i)$. The same argument goes for the truncated Student.

We test the idea for computing the expectation of truncated multivariate Student. We use a Gibbs sampler as a benchmark based on Robert [1995]'s sampler by adding a MH step to deal with u (see previous section). The Gibbs update is done after a change of variable that leaves the truncations independent. This can be shown to be more efficient. We allocate 100 times more computational time to the Gibbs sampler than the SMC.

In Figure 3.4 we see that after thinning one out of 1000 points the ACF and trace plots point to bad exploration of the target's support. This behavior shows that the convergence is too slow for the algorithm to be of practical use. On the other hand the SMC is still stable as is shown in the next section.

In addition of outperforming the Gibbs sampler for fairly moderate dimension, the SMC algorithm was found to be stable for approximating the expectation in dimensions up to 100.



(a) Trace plot (b) ACF

(c) Trace plot (d) ACF

Figure 3.4: Truncated Student sampling after thinning one out of every 1000, using a Gibbs sampler

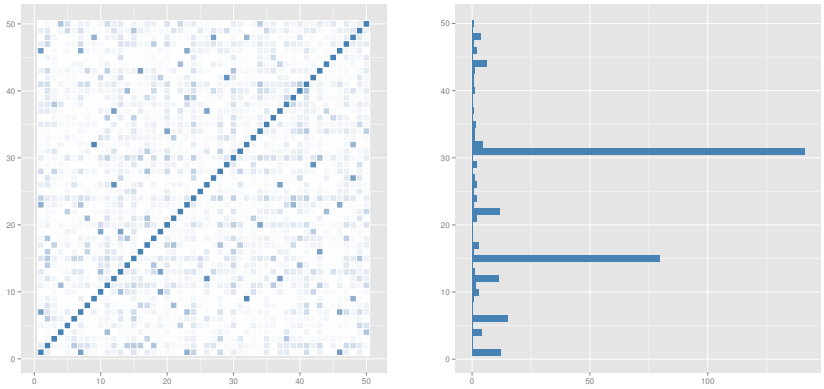
The data are generated as explained in the "Numerical results section", for dimension 50. The left panels are trace plots for two components. The right panels are the ACFs. Both are shown after a thinning of 1/1000. Both show a slow convergence, whereas we observe that SMC is stable.

3.6 Numerical results

3.6.1 Covariance simulation, tunning parameters

To build the covariance matrices we propose to use draws from a Cauchy distribution. We start by sampling a matrix X and a vector $\mathbf{a}$ from an independent Cauchy distribution $X_{ij} \sim \mathcal{C}(0, 0.01)$ and $a_i \sim \mathcal{C}(0, 0.01)$, then construct the covariance matrix as $\Sigma = X^t X$ and the truncation as $\mathbf{a} = (a_i)$. Because of the heavy tails the resulting correlation matrix (figure 3.5a) has many close to zero entries and some high correlations. The truncations also have some very high levels (figure 3.5b).

We find that this approach leads to more challenging covariances than those built by sampling the spectrum of the covariance matrix as proposed for instance in Christen et al. [2012].



(a) Correlations (b) Truncation

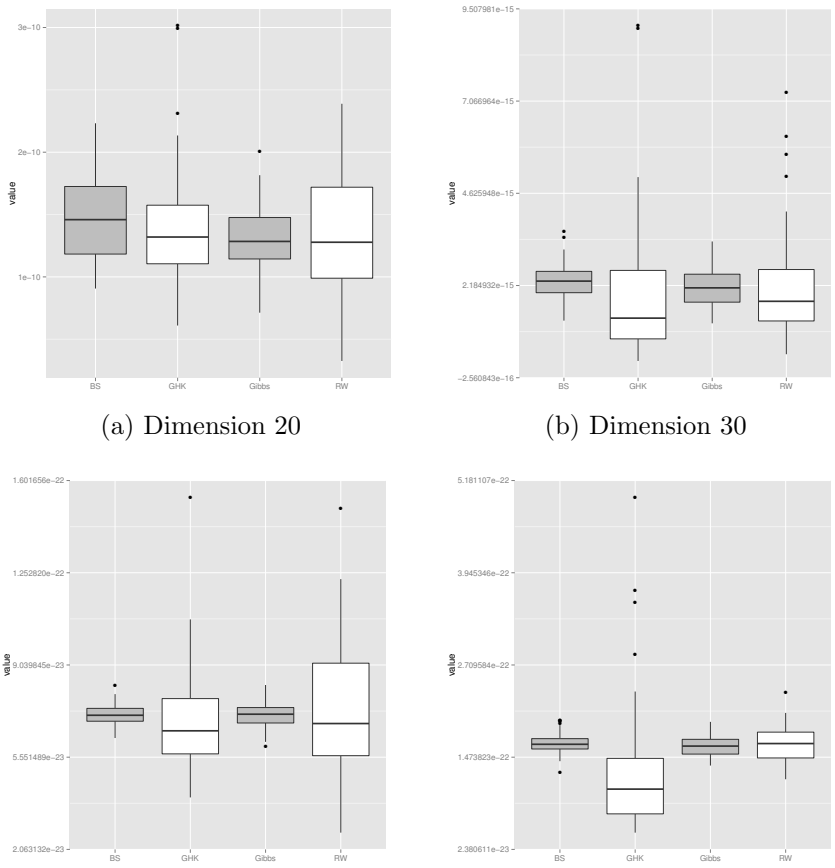
Figure 3.5: Generated correlations

Covariance matrices generated from random samples with heavy tails. The various dimension are simulated with the same algorithm and same seed, such that the small ones are subsets of the others. The left panel shows a heatmap of the correlation, the right panel the left trucation of the integral.

The tuning parameters of the algorithm are the threshold that tunes the number of steps of the MCMC kernel, and the targeted ESS under which we resample, $\alpha^\star$. The former is set to some small value (0.01) and has not much influence. The latter gives us a trade off between variance and computational cost. In our example we have found that $0.5M$ allows good approximation, however this value should be increased with the difficulty of the problem.

3.6.2 GHK for moderate dimensions

All our results are shown at constant computational cost: we repeat the algorithm in a first time to get their execution time, and we then scale them accordingly. In the above example (dimension 40) for instance the number of draws associated with the GHK algorithm is $1,065,399$. The number of particles of the SMC sampler with Gibbs Markov transition is 5217.



(a) Dimension 20 (b) Dimension 30

(c) Dimension 40 (d) Dimension 50

Figure 3.6: Estimates of Orthant probabilities for GHK compared to SMC with different moves.

The various dimensions are simulated as described in Section 3.6. Different moves are tested inside SMC. As we have already discussed GHK leads to a higher variance and outliers. Block sampling (BS) and Gibbs sampling (Gibbs) seem to outperform the overrelaxed random walk (RW). However all three stay stable until dimension 50.

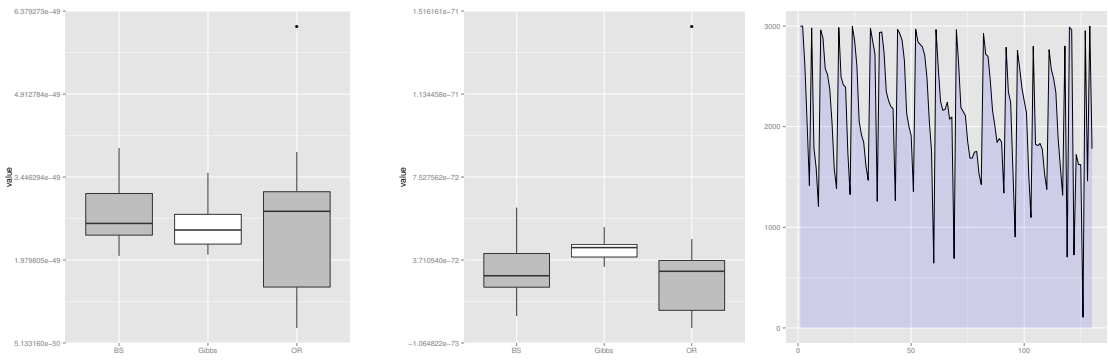
We find that in moderate dimension (~ 50) the GHK simulator breaks down in attempting to compute the probability of the orthant generated by our simulation scheme. This is all the more problematic as it gives an answer, and there is no way of checking its departure from the true value.

Another interesting aspect is the fact that the block sampling algorithm performs well in those dimensions. It is quicker than to move the particles in every dimension as is done with MCMC. The truncations that lead to a drop of probability are

close together because of the ordering, hence once the difficult dimensions have been "absorbed" it is less and less paramount to visit the past truncations.

3.6.3 High dimension orthant probabilities

In dimensions higher than $p = 70$, the covariance we simulate lead to integrals that cannot be treated with the GHK algorithm. In our simulations GHK always returned NaN values due to the low values of the weights. For the SMC an indicator of the good behavior of the algorithm can be seen in either its reproducibility and the fact that we do not encounter asymmetry (see Remark 3.3.1) as for the GHK in the first two Sections. Furthermore the ESS does not fall very low along the particles' draw (Figure 3.7c).



(a) Dimension 130 (b) Dimension 180 (c) Dimension 130

Figure 3.7: Estimates of Orthant probabilities $p = 130$ and $p = 180$

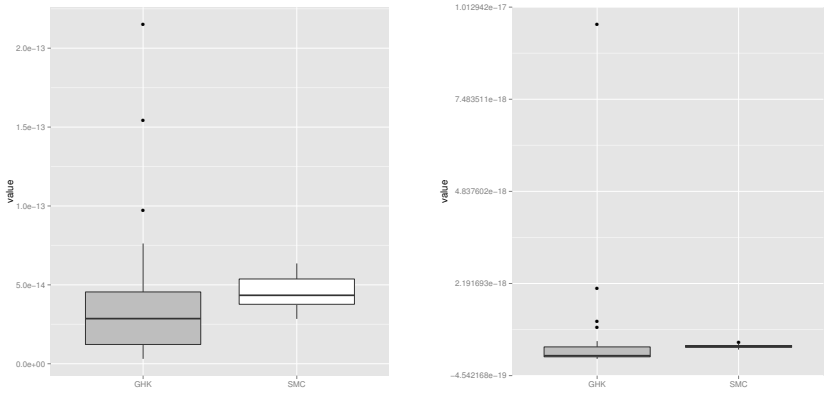
The various dimensions are simulated as described in Section 3.6. Different moves are tested inside SMC. As we have already discussed GHK leads to a higher variance and outliers. Gibbs sampling (Gibbs) seem to outperform the overrelaxed random walk (OR) and Block sampling (BS). However all three stay stable until dimension 180. The ESS for the Gibbs sampler is shown in panel c for a threshold of $0.5M$ and $M = 3000$. Despite some sudden drops it seems to be stable.

For those dimensions the Gibbs sampler performs best in terms of variance. However if one's goal is a fast algorithm, at the cost of higher variance the overrelaxation might be preferable at some point as the dimension of the target increases. The latter as a complexity smaller of one degree such that at constant computational cost it will have more and more particles allocated to it.

3.6.4 Student orthant probabilities

We use the same schemes as before to construct the covariance matrix and fix a degree of freedom of 3 in our experiments. As before we show an improvement as compared to previous algorithms. This improvements appears also for moderate dimensions. It seems that there is an important gain in considering the extended target.

As for the Gaussian case we find that the output of GHK is heavily skewed. It seems that it is not the case for our algorithm.



(a) Dimension 30 (b) Dimension 50

Figure 3.8: Estimates of Student orthant probabilities SMC vs GHK
Covariance matrices generated from random samples with heavy tails. We find that the SMC outperforms the GHK. As for the previous cases the GHK leads to some outliers.

3.6.5 Application to random utility models

Random utility models are an important area of research in Economics to model choice data [Train, 2009]. Consider an agent i confronted to J alternatives each giving utility $Y_{ij}^\star \; \forall j \in \{1, \cdots, J\}$ modeled by $Y_{ij}^\star = X_i\beta + u_{ij}$ with u_{ij} a Gaussian noise. Individual i chooses alternative j if $\{\forall k \neq j \quad Y_{ij}^\star > Y_{ik}^\star\}$. The likelihood is the probability of this set integrated over the unobserved alternatives. Hence the likelihood is given by:

$$L(Y_i = j|\Omega, \beta, X) = \mathbb{P}\left(\bigcap_{k \neq j}\{Y_{ij}^\star > Y_{ik}^\star\}\right)$$

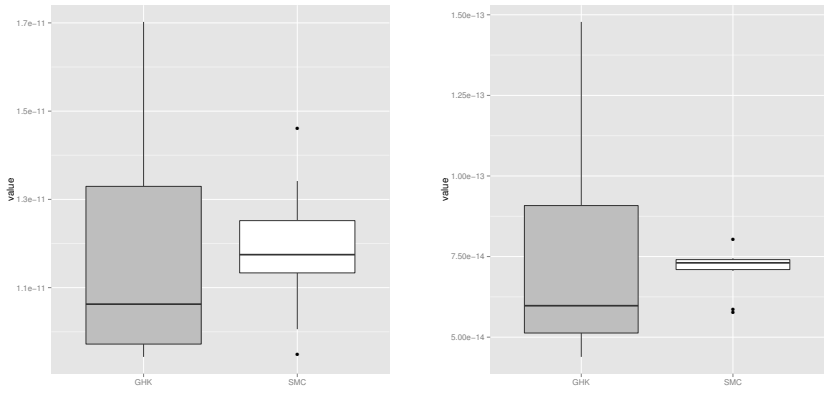
$$= \mathbb{P}\left(\bigcap_{k\neq j}\{(X_{ij} - X_{ik})\beta^\star > u_{ik} - u_{ij}\}\right)$$

where integration is taken over $u \sim \mathcal{N}(0, \Omega)$, where $u = (u_{ij})$. The above integral is an orthant probability of dimension $J - 1$. A yet more challenging case occurs in the presence of panel data. The latter corresponds to sequential choices of an individual in time. We denote those choices by the subscript t. We observe $(j_t)_{t<T}$ for every individual. Integration is now in dimension $T(J-1)$ and takes the form:

$$L(Y_{i,1:T} = j_{1:T}|\Omega, \beta, X) = \mathbb{P}\left(\bigcap_{t=1}^{T}\bigcap_{k\neq j_t}\{(X_{ij_t t} - X_{ikt})\beta^\star > u_{ikt} - u_{ij_t t}\}\right)$$

We take the covariance structure studied in Bursh-Supan et al. [1992]. The noise term is $u_{itk} = \alpha_{ik} + \eta_{ikt}$ where $\eta_{ikt} = \varrho_i \eta_{ikt-1} + \nu_{it}$, where (α_{ik}) are correlated amongst choices, so are ν_{it}. The terms are all Gaussian.

The dataset is simulated to allow for examples that are more complex, and of variable size. In the model presented above individuals are independent so that we present results in computing the integral for $n = 1$, and have already a big advantage of using our methodology.



(a) $J = 10$, $T = 10$ (b) $J = 10$, $T = 15$

Figure 3.9: Estimates of the likelihood of a multivariate Probit

Dataset: The Data is simulated using the covariance proposed in Bursh-Supan et al. [1992], the value of the parameter for which the likelihood is evaluated is taken at random.

Figure (3.9) shows, for two problems of different size, the gain in precision at constant computational cost. The improvement is substantial and increases with

the size of the problem. Taking $n > 1$ would only increase this effect as it would consist in taking products of such estimators.

The latter result suggests that we could use the likelihood for inference using either maximum likelihood or PMCMC. Computing the likelihood with lowest possible variance is also a key issue in finding the evidence in the most precise manner possible.

When comparing the two algorithms we set the number of particles of SMC to $M = 1000$, for the same computational cost we allocate $M = 881031$ to GHK. SMC has however still a lower variance. Regardless of computational time, the ability to compute precise integrals for a small number of particles can also be of importance. It is the case for instance in SMC^2 [Chopin et al., 2013b] where whole trajectories have to be kept in memory for several samplers.

One could extend these models to multivariate Probit with Student distribution and to multivariate logit models for more robustness. In this case we can use the algorithm based on the mixture representation built in Section 3.5.1.

3.7 Conclusion

We have shown empirically that the GHK algorithm collapses when the dimension of the problem increases (returning NaN values). In other cases, the distribution of estimates generated by GHK may have heavy tails (see also Remark 3.3.1). Theoretically for at least one covariance structure we have shown that the variance of the algorithm diverges exponentially fast with the dimension (Section 3.3). Our SMC algorithm seems to correct this behavior, and was found to be of practical use for many problems.

We have tested several kernels as part of the move step (Section 3.4.2). We advise the practitioners to use Gibbs sampling as the standard "go to" move step. However improvements in speed can be achieved for dimensions around 50 using only a partial update. In addition as the dimension increases one might want to use a method with lower complexity at the cost of having to repeat the move a bit more. In this case we recommend the use of an overrelaxed random walk Metropolis-Hastings.

We have shown that the same idea can be use for computing probabilities of mixtures of Gaussians. In addition we can use the weighted particles returned by the algorithm to compute other integrals (mean, variance, .etc). This approach

can outperform a classical Gibbs sampler when the dimension exceeds 20.

Appendix

3.A Proof of proposition 2.1

Proof: We have that for $0 < b < \infty$ the transition density $p^b(x, y)dy$, associated with the kernel $P^b(x, \mathrm{d}y)$ with respect to the Lebesgue measure, is lower bounded by a constant and the transition is continuous. This Markov chain is a ψ-irreducible on a compact support. Hence we can show that the whole support $[0, b]$ is small [Meyn and Tweedie, 2009].

Hence by theorem 16.1.2 to show V-Uniform ergodicity the transition must satisfy the drift condition:

$$\int p(x, y)V_e(y)dy \leq (1 - \beta)V_e(x) + c\mathbb{1}_{[0,b]}(x)$$

for $\beta > 0$, $c < \infty$ and a certain $V_e(x)$ with value in $[1, \infty)$. We take $V_e(x) = ex^2+1$, $e > 0$. In the following we check this condition.

The left hand side is given by $\mathbb{E}(X_t^2|X_{t-1} = x)$ for the above transition probability,

$$\mathbb{E}(X_t^2|X_{t-1} = x) = \varrho^2x^2 + 1 + \frac{\varrho\varphi(\varrho x) - b\varphi(b - \varrho x)}{\Phi(b - \varrho x) - \Phi(-\varrho x)}\varrho x$$

The ratio is continuous on the bounded set $[0, b]$, and can be bounded by a constant, such that the by taking $\beta = 1 - \varrho^2 > 0$ the drift condition is satisfied for a $c(e)$ depending on e.

In addition we can compute exactly the invariant measure. It is unique and given by the solution of:

$$\pi(y) = \int \frac{\varphi(y; \varrho x, 1)}{\Phi(a - \varrho x) - \Phi(-\varrho x)}\mathbb{1}_{[0,b]}(y)\pi(x)dx$$

The solution of the above equation is a truncated skew-Normal distribution,

$$\pi(dy) \propto \{\Phi(b-\varrho y) - \Phi(-\varrho y)\}\, \varphi(y; 0, 1-\varrho^2) \mathbb{1}_{[0,b]}(y) dy.$$

The moments of this distribution have been studied in Flecher et al. [2009], in particular note that $0 < \mathbb{V}_\pi(X) < \infty$.

Define $\psi : x \mapsto \log\left[\Phi(b-\varrho x) - \Phi(-\varrho x)\right]$, by theorem 17.0.1 [Meyn and Tweedie, 2009] to obtain a CLT for $\frac{1}{\sqrt{T}} \sum_{t=1}^{T} \psi(X_t)$ we must ensure that there exist a constant $e > 0$ such that $\psi^2(x) < V_e(x)$ on $[0, b]$. Such a constant can be found by noting that $\psi(x)$ is bounded as long as $b > 0$ and that V_e is strictly increasing of $e > 0$ with value on $(0, \infty)$. The value of e depends on b. We obtain the following convergence result,

$$\frac{1}{\sqrt{T}} \left(\sum_{t=1}^{T} \psi(X_t) - T\mathbb{E}_\pi(\psi(X)) \right) \rightsquigarrow \mathcal{N}\left(0, \mathbb{V}_\pi\left\{\psi(X)\right\} + \tau\right),$$

where the variance term is defined because ψ is bounded on $[0, b]$, and $\tau = 2\sum_{k=1}^{\infty} \mathrm{cov}(X_0, X_k)$. By taking the exponential and using the continuous mapping theorem (p.7 Van der Vaart [1998]) we get a log-normal limiting distribution

$$\left(\frac{\prod_{t=1}^{T} \left(\Phi(b-\varrho X_t) - \Phi(-\varrho X_t)\right)}{\exp\{T\mathbb{E}_\pi \log\left(\Phi(b-\varrho X) - \Phi(-\varrho X)\right)\}} \right)^{\frac{1}{\sqrt{T}}} \rightsquigarrow \mathcal{EN}\left(0, \mathbb{V}_\pi\left\{\psi(X)\right\} + \tau\right).$$

By Portmanteau's Lemma (p.6 Van der Vaart [1998]) for x^2 as a continuous and positive function,

$$\liminf_{T\to\infty} \mathbb{E}\left[\left(\frac{\prod_{t=1}^{T} \left(\Phi(b-\varrho X_t) - \Phi(-\varrho X_t)\right)^2}{\exp\{2T\mathbb{E}_\pi \log\left(\Phi(b-\varrho X) - \Phi(-\varrho X)\right)\}} \right)^{\frac{1}{\sqrt{T}}} \right] > \exp\left\{\mathbb{V}_\pi\left[\psi(X)\right] + \tau\right\}$$

$$\liminf_{T\to\infty} \mathbb{E}\left[\left(\frac{\prod_{t=1}^{T} \left(\Phi(b-\varrho X_t) - \Phi(-\varrho X_t)\right)^2}{\exp\{2T\mathbb{E}_\pi \log\left(\Phi(b-\varrho X) - \Phi(-\varrho X)\right)\}} \right) \right]^{\frac{1}{\sqrt{T}}} > \exp\left\{\mathbb{V}_\pi\left[\psi(X)\right] + \tau\right\}$$

where the last line is obtained by Jensen inequality. The denominator is the square of limit value of the normalizing constant under x^2-Uniform ergodicity that follows from the above statement. □

3.B Resampling

Algorithm 13 Systematic resampling, n particles

Input: Vector of weights w and vector x to sample from
Set $v \leftarrow nw$, $j \leftarrow 1$, $c = v_1$
Sample: Sample $U \sim \mathcal{U}_{[0,1]}$
for $k = 1, \cdots, n$ **do**
  while $c < u$ **do**
    Set $j \leftarrow j + 1$, $c \leftarrow c + v_j$
  end while
  Set: $\hat{x}_k \leftarrow x_j$, $u \leftarrow u + 1$
end for
return $\hat{x}$

3.C Variable Ordering

Algorithm 14 Variable Ordering

INIT: $i_1 = \arg\min_{1 \le k \le T} \Phi\left(\left[\frac{a_k}{\gamma_{kk}}, \frac{b_k}{\gamma_{kk}}\right]\right)$

$\eta_1 = \frac{1}{\Phi\left(\left[\frac{a_{i_1}}{\gamma_{i_1 i_1}}, \frac{b_{i_1}}{\gamma_{i_1 i_1}}\right]\right)} \int_{\left[\frac{a_{i_1}}{\gamma_{i_1 i_1}}, \frac{b_{i_1}}{\gamma_{i_1 i_1}}\right]} \eta\varphi(\eta)d\eta$

for $i \in \{2, \cdots, d\}$ **do**

STEP 1 $i_j = \arg\min_{j \le k \le T} \Phi\left(\left[\frac{1}{\tilde{\gamma}_{kk}}\left\{a_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}, \frac{1}{\tilde{\gamma}_{kk}}\left\{b_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}\right]\right)$

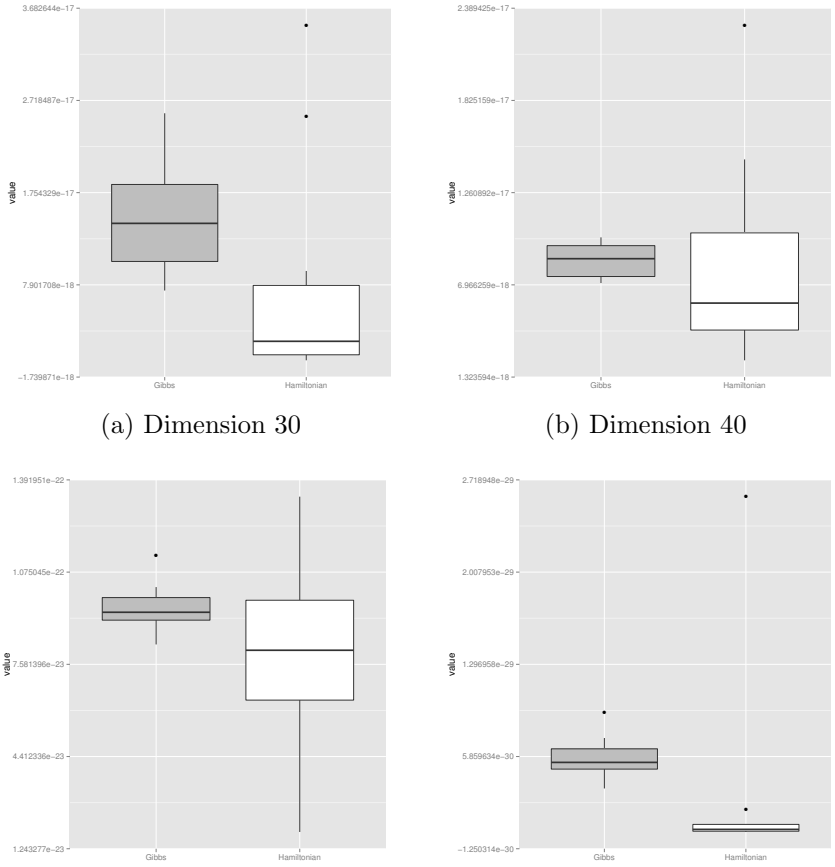
STEP 2

$$\eta_j = \frac{1}{\Phi\left(\left[\frac{1}{\tilde{\gamma}_{kk}}\left\{a_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}, \frac{1}{\tilde{\gamma}_{kk}}\left\{b_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}\right]\right)} \times \int_{\left[\frac{1}{\tilde{\gamma}_{kk}}\left\{a_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}, \frac{1}{\tilde{\gamma}_{kk}}\left\{b_k - \sum_{l=1}^{j-1} \tilde{\gamma}_{i_l k}\eta_k\right\}\right]} \eta\varphi(\eta)d\eta$$

end for
return $(i_1, \cdots, i_d)$

Where $\tilde{\gamma}$ is updated accordingly when the order is changed.

3.D Hamiltonian Monte Carlo



(a) Dimension 30 (b) Dimension 40

(c) Dimension 50 (d) Dimension 60

Figure 3.D.1: Estimates of Orthant probabilities Gibbs vs HMC

Covariance matrices generated from random samples with heavy tails. The various dimension are simulated with the same algorithm and same seed, such that the small ones are subsets of the others. The grey boxplot corresponds to the Gibbs sampler the white to the HMC. The Gibbs sampler seem to have smaller variance and no outliers.

4

Theoretical and computational aspects of PAC Bayesian ranking and scoring

This is joint work with Pierre Alquier, Nicolas Chopin and Feng Liang
Status: Published in *NIPS proceedings.*

4.1 Introduction

Bipartite ranking (scoring) amounts to rank (score) data from binary labels. An important problem in its own right, bipartite ranking is also an elegant way to formalise classification: once a score function has been estimated from the data, classification reduces to chooses a particular threshold, which determine to which class is assigned each data-point, according to whether its score is above or below that threshold. It is convenient to choose that threshold only once the score has been estimated, so as to get finer control of the false negative and false positive rates; this is easily achieved by plotting the ROC (Receiver operating characteristic) curve.

A standard optimality criterion for scoring is AUC (Area Under Curve), which measures the area under the ROC curve. AUC is appealing for at least two reasons. First, maximising AUC is equivalent to minimising the L_1 distance between the estimated score and the optimal score. Second, under mild conditions, Cortes and Mohri [2003] show that AUC for a score s equals the probability that $s(X^-) < s(X^+)$ for X^- (resp. X^+) a random draw from the negative (resp. positive class). Yan et al. [2003] observed AUC-based classification handles much better skewed classes (say the positive class is much larger than the other) than

standard classifiers, because it enforces a small score for all members of the negative class (again assuming the negative class is the smaller one).

One practical issue with AUC maximisation is that the empirical version of AUC is not a continuous function. One way to address this problem is to "convexify" this function, and study the properties of so-obtained estimators [Clémençon et al., 2008a]. We follow instead the PAC-Bayesian approach in this chapter, which consists of using a random estimator sampled from a pseudo-posterior distribution that penalises exponentially the (in our case) AUC risk. It is well known [see e.g. the monograph of Catoni, 2007] that the PAC-Bayesian approach comes with a set of powerful technical tools to establish non-asymptotic bounds; the first part of the chapter derive such bounds. A second advantage however of this approach, as we show in the second part of the chapter, is that it is amenable to powerful Bayesian computational tools, such as Sequential Monte Carlo and Expectation Propagation.

4.2 Theoretical bounds from the PAC-Bayesian Approach

4.2.1 Notations

The data $\mathcal{D}$ consist in the realisation of n IID (independent and identically distributed) pairs (X_i, Y_i) with distribution P, and taking values in $\mathbb{R}^d \times \{-1, 1\}$. Let $n_+ = \sum_{i=1}^n \mathbb{1}\{Y_i = +1\}$, $n_- = n - n_+$. For a score function $s : \mathbb{R}^d \to \mathbb{R}$, the AUC risk and its empirical counter-part may be defined as:

$$R(s) = \mathbb{P}_{(X,Y),(X',Y')\sim P}\left[\{s(X) - s(X')\}(Y - Y') < 0\right],$$
$$R_n(s) = \frac{1}{n(n-1)} \sum_{i \neq j} \mathbb{1}\left[\{s(X_i) - s(X_j)\}(Y_i - Y_j) < 0\right].$$

Let $\sigma(x) = \mathbb{E}(Y|X = x)$, $\bar{R} = R(\sigma)$ and $\bar{R}_n = R_n(\sigma)$. It is well known that σ is the score that minimise $R(s)$, i.e. $R(s) \geq \bar{R} = R(\sigma)$ for any score s.

The results of this section apply to the class of linear scores, $s_\theta(x) = \langle \theta, x \rangle$, where $\langle \theta, x \rangle = \theta^T x$ denotes the inner product. Abusing notations, let $R(\theta) = R(s_\theta)$, $R_n(\theta) = R_n(s_\theta)$, and, for a given prior density $\pi_\xi(\theta)$ that may depend on some hyperparameter $\xi \in \Xi$, define the Gibbs posterior density (or pseudo-posterior) as

$$\pi_{\xi,\gamma}(\theta|\mathcal{D}) := \frac{\pi_\xi(\theta) \exp\left\{-\gamma R_n(\theta)\right\}}{Z_{\xi,\gamma}(\mathcal{D})}, \quad Z_{\xi,\gamma}(\mathcal{D}) = \int_{R^d} \pi_\xi(\tilde{\theta}) \exp\left\{-\gamma R_n(\tilde{\theta})\right\} \mathrm{d}\tilde{\theta}$$

for $\gamma > 0$. Both the prior and posterior densities are defined with respect to the Lebesgue measure over $\mathbb{R}^d$.

4.2.2 Assumptions and general results

Our general results require the following assumptions.

Definition 4.2.1 *We say that Assumption* $\mathbf{Dens}(c)$ *is satisfied for* $c > 0$ *if*

$$\mathbb{P}(\langle X_1 - X_2, \theta \rangle \geq 0, \langle X_1 - X_2, \theta' \rangle \leq 0) \leq c\|\theta - \theta'\|$$

for any θ *and* $\theta' \in \mathbb{R}^d$ *such that* $\|\theta\| = \|\theta'\| = 1$.

This is a mild Assumption, which holds for instance as soon as $(X_1 - X_2)/\|X_1 - X_2\|$ admits a bounded probability density; see the appendix.

Definition 4.2.2 (Mammen & Tsybakov margin assumption) *We say that Assumption* $\mathbf{MA}(\kappa, C)$ *is satisfied for* $\kappa \in [1, +\infty]$ *and* $C \geq 1$ *if*

$$\mathbb{E}\left[(q^\theta_{1,2})^2\right] \leq C\left[R(\theta) - \overline{R}\right]^{\frac{1}{\kappa}}$$

where $q^\theta_{i,j} = \mathbb{1}\{\langle \theta, X_i - X_j \rangle (Y_i - Y_j) < 0\} - \mathbb{1}\{[\sigma(X_i) - \sigma(X_j)](Y_i - Y_j) < 0\} - R(\theta) + \overline{R}$.

This assumption was introduced for classification by Mammen and Tsybakov [1999], and used for ranking by Clémençon et al. [2008b] and Robbiano [2013] (see also a nice discussion in Lecué [2007]). The larger κ, the less restrictive $\mathbf{MA}(\kappa, C)$. In fact, $\mathbf{MA}(\infty, C)$ is always satisfied for $C = 4$. For a noiseless classification task (i.e. $\sigma(X_i)Y_i \geq 0$ almost surely), $\overline{R} = 0$,

$$\mathbb{E}((q^\theta_{1,2})^2) = \mathrm{Var}(q^\theta_{1,2}) = \mathbb{E}[\mathbb{1}\{\langle \theta, X_1 - X_2 \rangle (Y_i - Y_j) < 0\}] = R(\theta) - \overline{R}$$

and $\mathbf{MA}(1, 1)$ holds. More generally, $\mathbf{MA}(1, C)$ is satisfied as soon as the noise is small; see the discussion in Robiano 2013 (Proposition 5 p. 1256) for a formal statement. From now, we focus on either $\mathbf{MA}(1, C)$ or $\mathbf{MA}(\infty, C)$, $C \geq 1$. It is possible to prove convergence under $\mathbf{MA}(\kappa, 1)$ for a general $\kappa \geq 1$, but at the price of complications regarding the choice of γ; see Catoni [2007], Alquier [2008] and Robbiano [2013].

We use the classical PAC-Bayesian methodology initiated by McAllester [1998]; Shawe-Taylor and Williamson [1997] (see Alquier [2008]; Catoni [2007] for a complete survey and more recent advances) to get the following results. Proof of these and forthcoming results may be found in the appendix. Let $\mathcal{K}(\rho, \pi)$ denotes the Kullback-Liebler divergence, $\mathcal{K}(\rho, \pi) = \int \rho(\mathrm{d}\theta) \log\{\frac{\mathrm{d}\rho}{\mathrm{d}\pi}(\theta)\}$ if $\rho << \pi$, ∞ otherwise, and denote $\mathcal{M}^1_+$ the set of probability distributions $\rho(\mathrm{d}\theta)$.

Lemma 4.2.1 *Assume that* $\mathbf{MA}(1, C)$ *holds with* $C \geq 1$. *For any fixed* γ *with* $0 < \gamma \leq (n-1)/(8C)$, *for any* $\varepsilon > 0$, *with probability at least* $1 - \varepsilon$ *on the drawing of the data* $\mathcal{D}$,

$$\int R(\theta)\pi_{\xi,\gamma}(\theta|\mathcal{D})\mathrm{d}\theta - \overline{R} \leq 2 \inf_{\rho \in \mathcal{M}_+^1} \left\{ \int R(\theta)\rho(\mathrm{d}\theta) - \overline{R} + 2\frac{\mathcal{K}(\rho, \pi_\xi) + \log\left(\frac{4}{\varepsilon}\right)}{\gamma} \right\}.$$

Lemma 4.2.2 *Assume* $\boldsymbol{MA}(\infty, C)$ *with* $C \geq 1$. *For any fixed* γ *with* $0 < \gamma \leq (n-1)/8$, *for any* $\epsilon > 0$ *with probability* $1 - \epsilon$ *on the drawing of* $\mathcal{D}$,

$$\int R(\theta)\pi_{\xi,\gamma}(\theta|\mathcal{D})\mathrm{d}\theta - \bar{R} \leq \inf_{\rho \in \mathcal{M}_+^1} \left\{ \int R(\theta)\rho(d\theta) - \bar{R} + 2\frac{\mathcal{K}(\rho, \pi_\xi) + \log\frac{2}{\epsilon}}{\gamma} \right\} + \frac{16\gamma}{n-1}.$$

Both lemmas bound the expected risk excess, for a random estimator of θ generated from $\pi_{\xi,\gamma}(\theta|\mathcal{D})$.

4.2.3 Independent Gaussian Prior

We now specialise these results to the prior density $\pi_\xi(\theta) = \prod_{i=1}^d \varphi(\theta_i; 0, \vartheta)$, i.e. a product of independent Gaussian distributions $N(0, \vartheta)$; $\xi = \vartheta$ in this case.

Theorem 4.2.3 *Assume* $\mathbf{MA}(1, C)$, $C \geq 1$, $\mathbf{Dens}(c)$, $c > 0$, *and take* $\vartheta = \frac{2}{d}(1 + \frac{1}{n^2 d})$, $\gamma = (n-1)/8C$, *then there exists a constant* $\alpha = \alpha(c, C, d)$ *such that for any* $\epsilon > 0$, *with probability* $1 - \epsilon$,

$$\int R(\theta)\pi_\gamma(\theta|\mathcal{D})\mathrm{d}\theta - \bar{R} \leq 2 \inf_{\theta_0} \left\{ R(\theta_0) - \bar{R} \right\} + \alpha\frac{d\log(n) + \log\frac{4}{\epsilon}}{n-1}.$$

Theorem 4.2.4 *Assume* $\mathbf{MA}(\infty, C)$, $C \geq 1$, $\mathbf{Dens}(c)$ $c > 0$, *and take* $\vartheta = \frac{2}{d}(1 + \frac{1}{n^2 d})$, $\gamma = C\sqrt{dn\log(n)}$, *there exists a constant* $\alpha = \alpha(c, C, d)$ *such that for any* $\epsilon > 0$, *with probability* $1 - \epsilon$,

$$\int R(\theta)\pi_\gamma(\theta|\mathcal{D})\mathrm{d}\theta - \bar{R} \leq \inf_{\theta_0} \left\{ R(\theta_0) - \bar{R} \right\} + \alpha\frac{\sqrt{d\log(n)} + \log\frac{2}{\epsilon}}{\sqrt{n}}.$$

The proof of these results is provided in the appendix. It is known that, under $\mathbf{MA}(\kappa, C)$, the rate $(d/n)^{\frac{\kappa}{2\kappa-1}}$ is minimax-optimal for classification problems, see Lecué [2007]. Following Robbiano [2013] we conjecture that this rate is also optimal for ranking problems.

4.2.4 Spike and slab prior for feature selection

The independent Gaussian prior considered in the previous section is a natural choice, but it does not accommodate sparsity, that is, the possibility that only a small subset of the components of X_i actually determine the membership to either class. For sparse scenarios, one may use the spike and slab prior of Mitchell and Beauchamp [1988], George and McCulloch [1993b],

$$\pi_\xi(\theta) = \prod_{i=1}^{d} [p\varphi(\theta_i; 0, v_1) + (1-p)\varphi(\theta_i; 0, v_0)]$$

with $\xi = (p, v_0, v_1) \in [0,1] \times (\mathbb{R}^+)^2$, and $v_0 \ll v_1$, for which we obtain the following result. Note $\|\theta\|_0$ is the number of non-zero coordinates for $\theta \in \mathbb{R}^d$.

Theorem 4.2.5 *Assume* $\mathbf{MA}(1, C)$ *holds with* $C \geq 1$, $\mathbf{Dens}(c)$ *holds with* $c > 0$, *and take* $p = 1 - \exp(-1/d)$, $v_0 \leq 1/(2nd\log(d))$, *and* $\gamma = (n-1)/(8C)$. *Then there is a constant* $\alpha = \alpha(C, v_1, c)$ *such that for any* $\varepsilon > 0$, *with probability at least* $1 - \varepsilon$ *on the drawing of the data* $\mathcal{D}$,

$$\int R(\theta)\pi_\gamma(\mathrm{d}\theta|\mathcal{D}) - \overline{R} \leq 2 \inf_{\theta_0} \left\{ R(\theta_0) - \overline{R} + \alpha \frac{\|\theta_0\|_0 \log(nd) + \log\left(\frac{4}{\varepsilon}\right)}{2(n-1)} \right\}.$$

Compared to Theorem 4.2.3, the bound above increases logarithmically rather than linearly in d, and depends explicitly on $\|\theta\|_0$, the sparsity of θ. This suggests that the spike and slab prior should lead to better performance than the Gaussian prior in sparse scenarios. The rate $\|\theta\|_0 \log(d)/n$ is the same as the one obtained in sparse regression, see e.g. Bühlmann and van de Geer [2011].

Finally, note that if $v_0 \to 0$, we recover the more standard prior which assigns a point mass at zero for every component. However this leads to a pseudo-posterior which is a mixture of 2^d components that mix Dirac masses and continuous distributions, and thus which is more difficult to approximate (although see the related remark in Section 4.3.4 for Expectation-Propagation).

4.3 Practical implementation of the PAC-Bayesian approach

4.3.1 Choice of hyper-parameters

Theorems 4.2.3, 4.2.4, and 4.2.5 propose specific values for hyper-parameters γ and ξ, but these values depend on some unknown constant C. Two data-driven

ways to choose γ and ξ are (i) cross-validation (which we will use for γ), and (ii) (pseudo-)evidence maximisation (which we will use for ξ).

The latter may be justified from intermediate results of our proofs in the appendix, which provide an empirical bound on the expected risk:

$$\int R(\theta)\pi_{\xi,\gamma}(\theta|\mathcal{D})\mathrm{d}\theta - \bar{R} \leq \Psi_{\gamma,n} \inf_{\rho\in\mathcal{M}_+^1} \left(\int R_n(\theta)\rho(d\theta) - \bar{R}_n + \frac{\mathcal{K}(\rho,\pi) + \log\frac{2}{\epsilon}}{\gamma} \right)$$

with $\Psi_{\gamma,n} \leq 2$. The right-hand side is minimised at $\rho(\mathrm{d}\theta) = \pi_{\xi,\gamma}(\theta|\mathcal{D})\mathrm{d}\theta$, and the so-obtained bound is $-\Psi_{\gamma,n}\log(Z_{\xi,\gamma}(\mathcal{D}))/\gamma$ plus constants. Minimising the upper bound with respect to hyperparameter ξ is therefore equivalent to maximising $\log Z_{\xi,\gamma}(\mathcal{D})$ with respect to ξ. This is of course akin to the empirical Bayes approach that is commonly used in probabilistic machine learning. Regarding γ the minimization is more cumbersome because the dependence with the $\log(2/\epsilon)$ term and $\Psi_{n,\gamma}$, which is why we recommend cross-validation instead.

It seems noteworthy that, beside Alquier and Biau [2013], very few papers discuss the practical implementation of PAC-Bayes, beyond some brief mention of MCMC (Markov chain Monte Carlo). However, estimating the normalising constant of a target density simulated with MCMC is notoriously difficult. In addition, even if one decides to fix the hyperparameters to some arbitrary value, MCMC may become slow and difficult to calibrate if the dimension of the sampling space becomes large. This is particularly true if the target does not (as in our case) have some specific structure that makes it possible to implement Gibbs sampling. The two next sections discuss two efficient approaches that make it possible to approximate both the pseudo-posterior $\pi_{\xi,\gamma}(\theta|\mathcal{D})$ and its normalising constant, and also to perform cross-validation with little overhead.

4.3.2 Sequential Monte Carlo

Given the particular structure of the pseudo-posterior $\pi_{\xi,\gamma}(\theta|\mathcal{D})$, a natural approach to simulate from $\pi_{\xi,\gamma}(\theta|\mathcal{D})$ is to use tempering SMC [Sequential Monte Carlo Del Moral et al., 2006b] that is, define a certain sequence $\gamma_0 = 0 < \gamma_1 < \ldots < \gamma_T$, start by sampling from the prior $\pi_\xi(\theta)$, then applies successive importance sampling steps, from $\pi_{\xi,\gamma_{t-1}}(\theta|\mathcal{D})$ to $\pi_{\xi,\gamma_t}(\theta|\mathcal{D})$, leading to importance weights proportional to:

$$\frac{\pi_{\xi,\gamma_t}(\theta|\mathcal{D})}{\pi_{\xi,\gamma_{t-1}}(\theta|\mathcal{D})} \propto \exp\left\{-(\gamma_t - \gamma_{t-1})R_n(\theta)\right\}.$$

When the importance weights become too skewed, one rejuvenates the particles through a resampling step (draw particles randomly with replacement, with probability proportional to the weights) and a move step (move particles according to a certain MCMC kernel).

One big advantage of SMC is that it is very easy to make it fully adaptive. For the choice of the successive γ_t, we follow Jasra et al. [2007] in solving numerically (5.18) in order to impose that the Effective sample size has a fixed value. This ensures that the degeneracy of the weights always remain under a certain threshold. For the MCMC kernel, we use a Gaussian random walk Metropolis step, calibrated on the covariance matrix of the resampled particles. See Algorithm 19 for a summary.

Algorithm 15 Tempering SMC

Input N (number of particles), $\tau \in (0,1)$ (ESS threshold), $\kappa > 0$ (random walk tuning parameter)

Init. Sample $\theta_0^i \sim \pi_\xi(\theta)$ for $i = 1$ to N, set $t \leftarrow 1$, $\gamma_0 = 0$, $Z_0 = 1$.

Loop a. Solve in γ_t the equation

$$\frac{\{\sum_{i=1}^N w_t(\theta_{t-1}^i)\}^2}{\sum_{i=1}^N \{w_t(\theta_{t-1}^i))^2\}} = \tau N, \quad w_t(\theta) = \exp[-(\gamma_t - \gamma_{t-1})R_n(\theta)] \tag{4.1}$$

using bisection search. If $\gamma_t \geq \gamma_T$, set $Z_T = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$, and stop.

b. Resample: for $i = 1$ to N, draw A_t^i in $1, \ldots, N$ so that $\mathbb{P}(A_t^i = j) = w_t(\theta_{t-1}^j) / \sum_{k=1}^N w_t(\theta_{t-1}^k)$; see Algorithm 1 in the appendix.

c. Sample $\theta_t^i \sim M_t(\theta_{t-1}^{A_t^i}, \mathrm{d}\theta)$ for $i = 1$ to N where M_t is a MCMC kernel that leaves invariant π_t; see Algorithm 3 in the appendix for an instance of such a MCMC kernel, which takes as an input $S = \kappa\hat{\Sigma}$, where $\hat{\Sigma}$ is the covariance matrix of the $\theta_{t-1}^{A_t^i}$.

d. Set $Z_t = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$.

In our context, tempering SMC brings two extra advantages: it makes it possible to obtain samples from $\pi_{\xi,\gamma}(\theta|\mathcal{D})$ for a whole range of values of γ, rather than a single value. And it provides an approximation of $Z_{\xi,\gamma}(\mathcal{D})$ for the same range of γ values, through the quantity Z_t defined in Algorithm 19.

4.3.3 Expectation-Propagation (Gaussian prior)

The SMC sampler outlined in the previous section works fairly well, and we will use it as gold standard in our simulations. However, as any other Monte Carlo method, it may be too slow for large datasets. We now turn our attention to EP [Expectation-Propagation Minka, 2001a], a general framework to derive fast approximations to target distributions (and their normalising constants).

First note that the pseudo-posterior may be rewritten as:

$$\pi_{\xi,\gamma}(\theta|\mathcal{D}) = \frac{1}{Z_{\xi,\gamma}(\mathcal{D})}\pi_\xi(\theta) \times \prod_{i,j} f_{ij}(\theta), \quad f_{ij}(\theta) = \exp\left[-\gamma' \mathbb{1}\{\langle \theta, X_i - X_j\rangle < 0\}\right]$$

where $\gamma' = \gamma/n_+n_-$, and the product is over all (i,j) such that $Y_i = 1$, $Y_j = -1$. EP generates an approximation of this target distribution based on the same factorisation:

$$q(\theta) \propto q_0(\theta) \prod_{i,j} q_{ij}(\theta), \quad q_{ij}(\theta) = \exp\{-\frac{1}{2}\theta^T Q_{ij}\theta + r_{ij}^T\theta\}.$$

We consider in the section the case where the prior is Gaussian, as in Section 4.2.3. Then one may set $q_0(\theta) = \pi_\xi(\theta)$. The approximating factors are unnormalised Gaussian densities (under a natural parametrisation), leading to an overall approximation that is also Gaussian, but other types of exponential family parametrisations may be considered; see next section and Seeger [2005a]. EP updates iteratively each site q_{ij} (that is, it updates the parameters Q_{ij} and r_{ij}), conditional on all the sites, by matching the moments of q with those of the hybrid distribution

$$h_{ij}(\theta) \propto q(\theta)\frac{f_{ij}(\theta)}{q_{ij}(\theta)} \propto q_0(\theta) f_{ij}(\theta) \prod_{(k,l)\neq(i,j)} q_{kl}(\theta)$$

where again the product is over all (k,l) such that $Y_k = 1$, $Y_l = -1$, and $(k,l) \neq (i,j)$.

We refer to the appendix for a precise algorithmic description of our EP implementation. We highlight the following points. First, the site update is particularly simple in our case:

$$h_{ij}(\theta) \propto \exp\{\theta^T r_{ij}^h - \frac{1}{2}\theta^T Q_{ij}^h \theta\} \exp\left[-\gamma' \mathbb{1}\{\langle \theta, X_i - X_j\rangle < 0\}\right],$$

with $r_{ij}^h = \sum_{(k,l)\neq(i,j)} r_{kl}$, $Q_{ij}^h = \sum_{(k,l)\neq(i,j)} Q_{kl}$, which may be interpreted as: θ conditional on $T(\theta) = \langle \theta, X_i - X_j\rangle$ has a $d-1$-dimensional Gaussian distribution, and the distribution of $T(\theta)$ is that of a one-dimensional Gaussian penalised by a step function. The two first moments of this particular hybrid may therefore be

computed exactly, and in $\mathcal{O}(d^2)$ time, as explained in the appendix. The updates can be performed efficiently using the fact that the linear combination $(X_i - X_j)\theta$ is a one dimensional Gaussian. For our numerical experiment we used a parallel version of EP Van Gerven et al. [2010]. The complexity of our EP implementation is $\mathcal{O}(n_+ n_- d^2 + d^3)$.

Second, EP offers at no extra cost an approximation of the normalising constant $Z_{\xi,\gamma}(\mathcal{D})$ of the target $\pi_{\xi,\gamma}(\theta|\mathcal{D})$; in fact, one may even obtain derivatives of this approximated quantity with respect to hyper-parameters. See again the appendix for more details.

Third, in the EP framework, cross-validation may be interpreted as dropping all the factors q_{ij} that depend on a given data-point X_i in the global approximation q. This makes it possible to implement cross-validation at little extra cost [Opper and Winther, 2000].

4.3.4 Expectation-Propagation (spike and slab prior)

To adapt our EP algorithm to the spike and slab prior of Section 4.2.4, we introduce latent variables $Z_k = 0/1$ which "choose" for each component θ_k whether it comes from a slab, or from a spike, and we consider the joint target

$$\pi_{\xi,\gamma}(\theta, z|\mathcal{D}) \propto \left\{ \prod_{k=1}^{d} \mathcal{B}(z_k; p)\mathcal{N}(\theta_k; 0, v_{z_k}) \right\} \exp\left[-\frac{\gamma}{n_+ n_-} \sum_{ij} \mathbb{1}\{\langle \theta, X_i - X_j \rangle > 0\} \right].$$

On top of the $n_+ n_-$ Gaussian sites defined in the previous section, we add a product of d sites to approximate the prior. Following Hernandez-Lobato et al. [2013], we use

$$q_k(\theta_k, z_k) = \exp\left\{ z_k \log\left(\frac{p_k}{1 - p_k} \right) - \frac{1}{2}\theta_k^2 u_k + v_k \theta_k \right\}$$

that is a (un-normalised) product of an independent Bernoulli distribution for z_k, times a Gaussian distribution for θ_k. Again that the site update is fairly straightforward, and may be implemented in $\mathcal{O}(d^2)$ time. See the appendix for more details. Another advantage of this formulation is that we obtain a Bernoulli approximation of the marginal pseudo-posterior $\pi_{\xi,\gamma}(z_i = 1|\mathcal{D})$ to use in feature selection. Interestingly taking v_0 to be exactly zero also yield stable results corresponding to the case where the spike is a Dirac mass.

4.4 Extension to non-linear scores

To extend our methodology to non-linear score functions, we consider the pseudo-posterior

$$\pi_{\xi,\gamma}(\mathrm{d}s|\mathcal{D}) \propto \pi_\xi(\mathrm{d}s) \exp\left\{-\frac{\gamma}{n_+ n_-} \sum_{i\in\mathcal{D}_+, j\in\mathcal{D}_-} \mathbb{1}\{s(X_i) - s(X_j) > 0\}\right\}$$

where $\pi_\xi(\mathrm{d}s)$ is some prior probability measure with respect to an infinite-dimensional functional class. Let $s_i = s(X_i)$, $s_{1:n} = (s_1, \ldots, s_n) \in \mathbb{R}^n$, and assume that $\pi_\xi(\mathrm{d}s)$ is a GP (Gaussian process) associated to some kernel $k_\xi(x, x')$, then using a standard trick in the GP literature [Rasmussen and Williams, 2006], one may derive the marginal (posterior) density (with respect to the n-dimensional Lebesgue measure) of $s_{1:n}$ as

$$\pi_{\xi,\gamma}(s_{1:n}|\mathcal{D}) \propto \mathcal{N}_d\left(s_{1:n}; 0, K_\xi\right) \exp\left\{-\frac{\gamma}{n_+ n_-} \sum_{i\in\mathcal{D}_+, j\in\mathcal{D}_-} \mathbb{1}\{s_i - s_j > 0\}\right\}$$

where $\mathcal{N}_d\left(s_{1:n}; 0, K_\xi\right)$ denotes the probability density of the $\mathcal{N}(0, K_\xi)$ distribution, and K_ξ is the $n \times n$ matrix $\left(k_\xi(X_i, X_j)\right)_{i,j=1}^n$.

This marginal pseudo-posterior retains essentially the structure of the pseudo-posterior $\pi_{\xi,\gamma}(\theta|\mathcal{D})$ for linear scores, except that the "parameter" $s_{1:n}$ is now of dimension n. We can apply straightforwardly the SMC sampler of Section 4.B.1, and the EP algorithm of 4.B.2, to this new target distribution. In fact, for the EP implementation, the particular simple structure of a single site:

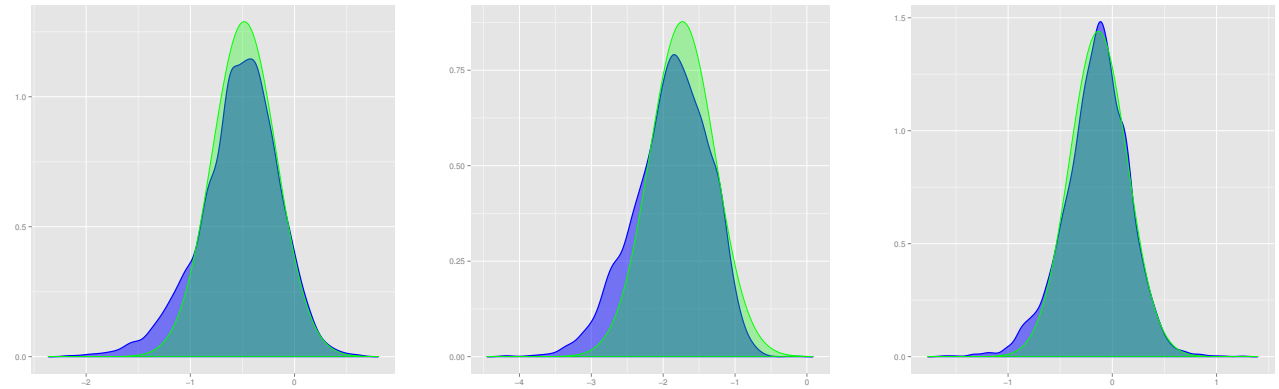
$$\exp\left[-\gamma' \mathbb{1}\{s_i - s_j > 0\}\right]$$

makes it possible to implement a site update in $\mathcal{O}(1)$ time, leading to an overall complexity $\mathcal{O}(n_+ n_- + n^3)$ for the EP algorithm.

Theoretical results for this approach could be obtained by applying lemmas from e.g. van der Vaart and van Zanten [2009], but we leave this for future study.

4.5 Numerical Illustration

Figure 1 compares the EP approximation with the output of our SMC sampler, on the well-known Pima Indians dataset and a Gaussian prior. Marginal first and second order moments essentially match; see the appendix for further details. The subsequent results are obtained with EP.



(a) θ_1 (b) θ_2 (c) θ_3

Figure 4.1: EP Approximation (green), compared to SMC (blue) of the marginal posterior of the first three coefficients, for Pima dataset (see the appendix for additional analysis).

We now compare our PAC-Bayesian approach (computed with EP) with Bayesian logistic regression (to deal with non-identifiable cases), and with the rankboost algorithm [Freund et al., 2003] on different datasets[1]; note that Cortes and Mohri [2003] showed that the function optimised by rankbook is AUC.

As mentioned in Section 4.B, we set the prior hyperparameters by maximizing the evidence, and we use cross-validation to choose γ. To ensure convergence of EP, when dealing with difficult sites, we use damping [Seeger, 2005a]. The GP version of the algorithm is based on a squared exponential kernel. Table 5.2 summarises the results; balance refers to the size of the smaller class in the data (recall that the AUC criterion is particularly relevant for unbalanced classification tasks), EP-AUC (resp. GPEP-AUC) refers to the EP approximation of the pseudo-posterior based on our Gaussian prior (resp. Gaussian process prior). See also Figure 2 for ROC curve comparisons, and Table 2 in the appendix for a CPU time comparison.

Note how the GP approach performs better for the colon data, where the number of covariates (2000) is very large, but the number of observations is only 40. It seems also that EP gives a better approximation in this case because of the lower dimensionality of the pseudo-posterior (Figure 4.2b).

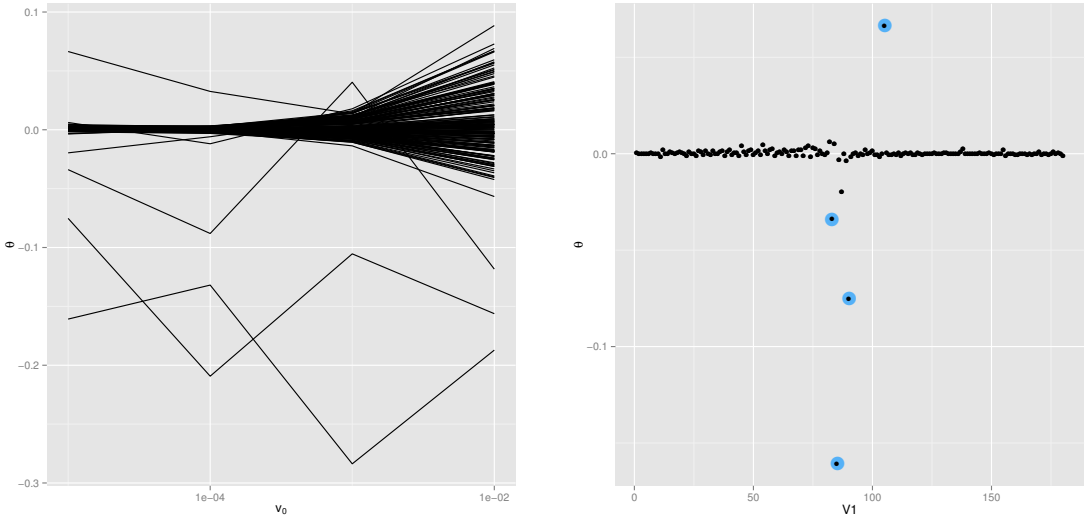
Finally, we also investigate feature selection for the DNA dataset (180 covariates) using a spike and slab prior. The regularization plot (4.3a) shows how certain coefficients shrink to zero as the spike's variance v_0 goes to zero, allowing for some

[1] All available at http://archive.ics.uci.edu/ml/

Dataset	Covariates	Balance	EP-AUC	GPEP-AUC	Logit	Rankboost
Pima	7	34%	0.8617	0.8557	**0.8646**	0.8224
Credit	60	28%	**0.7952**	0.7922	0.7561	0.788
DNA	180	22%	**0.9814**	0.9812	0.9696	**0.9814**
SPECTF	22	50%	0.8684	0.8545	**0.8715**	0.8684
Colon	2000	40%	0.7034	**0.75**	0.73	0.5935
Glass	10	1%	**0.9843**	0.9629	0.9029	0.9436

Table 4.1: Comparison of AUC.
The Glass dataset has originally more than two classes. We compare the "silicon" class against all others.

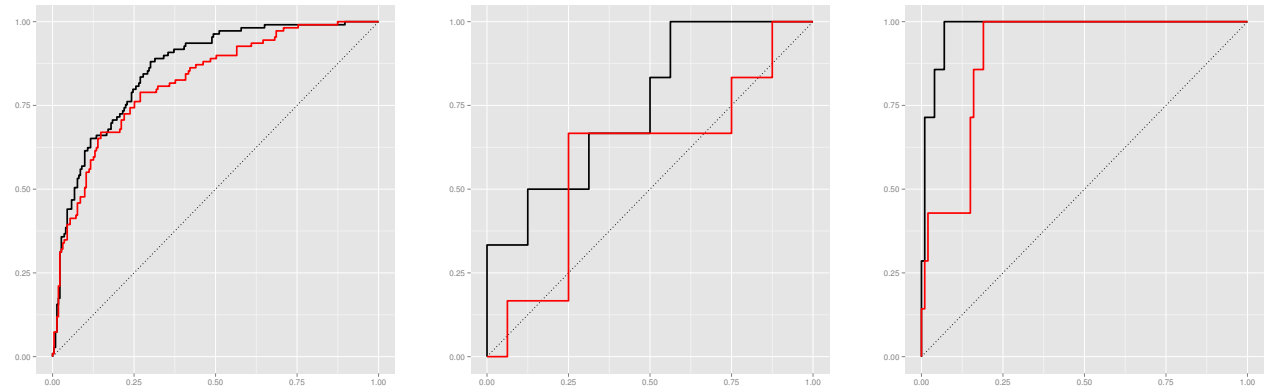
sparsity. The aim of a positive variance in the spike is to absorb negligible effects into it [Ročková and George, 2013]. We observe this effect on figure 4.3a where one of the covariates becomes positive when v_0 decreases.



(a) Regularization plot

(b) Estimate

Figure 4.3: Regularization plot for $v_0 \in \left[10^{-6}, 0.1\right]$ and estimation for $v_0 = 10^{-6}$ for DNA dataset; blue circles denote posterior probabilities ≥ 0.5.



(a) Rankboost vs EP-AUC on Pima

(b) Rankboost vs GPEP-AUC on Colon

(c) Logistic vs EP-AUC on Glass

Figure 4.2: Some ROC curves associated to the example described in a more systematic manner in table 5.2. In black is always the PAC version.

4.6 Conclusion

The combination of the PAC-Bayesian theory and Expectation-Propagation leads to fast and efficient AUC classification algorithms, as observed on a variety of datasets, some of them very unbalanced. Future work may include extending our approach to more general ranking problems (e.g. multi-class), establishing non-asymptotic bounds in the nonparametric case, and reducing the CPU time by considering only a subset of all the pairs of datapoints.

Appendix

4.A PAC-Bayes bounds for linear scores

4.A.1 Sufficient condition for Dens(c)

A simple sufficient condition for Dens(c) to hold is that $(X_1 - X_2)/\|X_1 - X_2\|$ admits a probability density with respect to the spherical measure of dimension $d-1$ which is bounded above by B. Then

$$\begin{aligned}\mathbb{P}(\langle X_1 - X_2, \theta\rangle \geq 0, \langle X_1 - X_2, \theta'\rangle \leq 0) &\leq B\frac{\arccos\left(\langle\theta, \theta'\rangle\right)}{2\pi} \\ &\leq \frac{B}{2\pi}\sqrt{5 - 5\left\langle\theta, \theta'\right\rangle} \\ &= \frac{B}{2\pi}\sqrt{\frac{5}{2}}\|\theta - \theta'\|.\end{aligned}$$

4.A.2 Proof of Lemma 2.1

In order to prove Lemma 2.1 we need the following Bernstein inequality.

Proposition 4.A.1 (Bernstein's inequality for U-statistics) *For any $\gamma > 0$, for any $\theta \in \mathbb{R}^d$,*

$$\mathbb{E}\exp[\gamma|R_n(\theta) - \overline{R}_n - R(\theta) + \overline{R}|] \leq 2\exp\left[\frac{\frac{\gamma^2}{n-1}\mathbb{E}((q^\theta_{1,2})^2)}{\left(1 - \frac{4\gamma}{n-1}\right)}\right].$$

Proof of Proposition 4.A.1. Fix θ. Remember that

$$q^\theta_{i,j} = \mathbf{1}\{\langle\theta, X_i - X_j\rangle (Y_i - Y_j) < 0\} - \mathbf{1}\{[\sigma(X_i) - \sigma(X_j)](Y_i - Y_j) < 0\} - R(\theta) + \overline{R}$$

so that

$$U_n := R_n(\theta) - \overline{R}_n - R(\theta) + \overline{R} = \frac{1}{n(n-1)}\sum_{i\neq j} q^\theta_{i,j}.$$

First, note that

$$\mathbb{E}\exp[\gamma|U_n|] \leq \mathbb{E}\exp[\gamma U_n] + \mathbb{E}\exp[\gamma(-U_n)].$$

We will only upper bound the first term in the r.h.s., as the upper bound for the second term may be obtained exactly in the same way (just replace $q^\theta_{i,j}$ by $-q^\theta_{i,j}$). Now, use Hoeffding's decomposition Hoeffding [1948]: this is the technique used by Hoeffding to prove inequalities on U-statistics. Hoeffding proved that

$$U_n = \frac{1}{n!}\sum_\pi \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)}$$

where the sum is taken over all the permutations π of $\{1,\dots,n\}$. Jensen's inequality leads to

$$\begin{aligned}\mathbb{E}\exp[\gamma U_n] &= \mathbb{E}\exp\left[\gamma \frac{1}{n!}\sum_\pi \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)}\right] \\ &\leq \frac{1}{n!}\sum_\pi \mathbb{E}\exp\left[\frac{\gamma}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)}\right].\end{aligned}$$

We now use, for each of the terms in the sum, Massart's version of Bernstein's inequality Massart [2007] (ineq. (2.21) in Chapter 2, the assumption is checked by $q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)} \in [-2,2]$ so $\mathbb{E}((q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)})^k) \leq \mathbb{E}((q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)})^2)2^{k-2}$). We obtain:

$$\mathbb{E}\exp\left[\frac{\gamma}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)}\right] \leq \exp\left[\frac{\mathbb{E}((q^\theta_{\pi(1),\pi(1+\lfloor \frac{n}{2} \rfloor)})^2)\frac{\gamma^2}{\lfloor \frac{n}{2} \rfloor}}{2\left(1-2\frac{\gamma}{\lfloor \frac{n}{2} \rfloor}\right)}\right].$$

First, note that we have the inequality $\lfloor \frac{n}{2} \rfloor \geq (n-1)/2$. Then, remark that as the pairs (X_i, Y_i) are iid, we have $\mathbb{E}((q^\theta_{\pi(1),\pi(1+\lfloor \frac{n}{2} \rfloor)})^2) = \mathbb{E}((q^\theta_{1,2})^2)$ so we have a simpler inequality

$$\mathbb{E}\exp\left[\frac{\gamma}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q^\theta_{\pi(i),\pi(i+\lfloor \frac{n}{2} \rfloor)}\right] \leq \exp\left[\frac{\mathbb{E}((q^\theta_{1,2})^2)\frac{\gamma^2}{n-1}}{\left(1-\frac{4\gamma}{n-1}\right)}\right].$$

This ends the proof of the proposition. □

The following proposition is also of use in the proof of lemma 2.1.

Proposition 4.A.2 *For any measure $\rho \in \mathcal{M}^1_+(\Theta)$ and any measurable function $h : \theta \to \mathbb{R}$ such that $\int \exp(h(\theta))\pi(\mathrm{d}\theta) < \infty$, we have*

$$\log\left(\int \exp(h(\theta))\pi(\theta)\right) = \sup_{\rho \in \mathcal{M}^1_+}\left(\int h(\theta)\rho(\mathrm{d}\theta) - \mathcal{K}(\rho,\pi)\right).$$

In addition if h is bounded by above on the support of π the supremum is reached for the Gibbs distribution,

$$\rho(\mathrm{d}\theta) \propto \exp\left(h(\theta)\right)\pi(\mathrm{d}\theta).$$

Proof: e.g. Catoni [2007]. □

Proof of Lemma 2.1 From the proof of Proposition 4.A.1, and using the shorthand q_θ for $q^\theta_{1,2}$, we deduce

$$\mathbb{E}\left[\exp\{\rho\left(\gamma(R_n(\theta) - \bar{R}_n - R(\theta) + \bar{R})\right\} + \eta(\theta))\right] \leq \exp\left(\frac{\gamma^2}{n-1}\frac{\rho\left(\mathbb{E}q_\theta^2\right)}{(1-4\frac{\gamma}{n-1})} + \rho\left(\eta(\theta)\right)\right). \tag{4.2}$$

Using proposition 4.A.2, and the fact that $e^x \geq \mathbb{1}\{x \geq 0\}$ we have that

$$\begin{aligned}
&\mathbb{P}\{\sup_{\rho\in\mathcal{M}_+^1(\Theta)} \rho\left(\gamma(R_n(\theta) - \bar{R}_n - R(\theta) + \bar{R}) - \eta(\theta)\right) - \mathcal{K}(\rho,\pi) \geq 0\} \\
&\leq \mathbb{E}\left(\pi\{\exp\{\rho\left(\gamma(R_n(\theta) - \bar{R}_n - R(\theta) + \bar{R}) - \eta(\theta)\right\}\right)\} \\
&= \pi\left(\mathbb{E}\{\exp\{\rho\left(\gamma(R_n(\theta) - \bar{R}_n - R(\theta) + \bar{R}) - \eta(\theta)\right\}\right)\} \quad \text{, by Fubini} \\
&\leq \pi\left\{\exp\left(\frac{\gamma^2\rho(\mathbb{E}q_\theta^2)}{(n-1)(1-\frac{4\gamma}{n-1})} - \rho(\eta(\theta))\right)\right\} \quad \text{, using (4.2).}
\end{aligned}$$

In the following we take $\eta(\theta) = \log\frac{1}{\epsilon} + \frac{\gamma^2}{n-1}\frac{\rho(\mathbb{E}q_\theta^2)}{(1-4\frac{\gamma}{n-1})}$ leading to the following result with probability at least $1-\epsilon$, $\forall\rho \in \mathcal{M}_+^1(\Theta)$:

$$\rho(R_n(\theta)) - \bar{R}_n \leq \rho(R(\theta)) - \bar{R} + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma} + \frac{\gamma}{n-1}\frac{\rho(\mathbb{E}q_\theta^2)}{(1-4\frac{\gamma}{n-1})}. \tag{4.3}$$

Under $\mathbf{MA}(1, C)$ we can write:

$$\rho(R_n(\theta)) - \bar{R}_n \leq \left(1 + \frac{\gamma C}{n-1}\frac{1}{(1-\frac{4}{n-1})}\right)\left(\rho(R(\theta)) - \bar{R}\right) + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma}.$$

Using Bernstein's inequality in the symmetric case, with probability $1-\epsilon$ we can assert that:

$$\left(1 - \frac{\gamma C}{n-1}\frac{1}{(1-\gamma\frac{4}{n-1})}\right)\left(\rho(R(\theta)) - \bar{R}\right) \leq \rho(R_n(\theta)) - \bar{R}_n + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma}.$$

The latter is true in particular for $\rho = \pi(\theta|\mathcal{S})$, the Gibbs posterior:

$$\left(1 - \frac{\gamma C}{n-1}\frac{1}{(1-\gamma\frac{4}{n-1})}\right)\left(\int_\Theta R(\theta)\pi_\gamma(d\theta|\mathcal{D}) - \bar{R}\right) \leq \inf_{\rho\in\mathcal{M}_+^1}\left\{\rho(R_n(\theta)) - \bar{R}_n + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma}\right\}.$$

Making use of equation (4.3) and the fact that $\gamma \leq (n-1)/8C$ we have with probability $1-2\epsilon$:

$$\left(\int_\Theta R_n(\theta)\pi_\gamma(d\theta|\mathcal{D}) - \bar{R}_n\right) \leq 2\inf_{\rho\in\mathcal{M}_+^1}\left(\rho(R(\theta)) - \bar{R} + 2\frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma}\right). \qquad \square$$

Lemma 2.1 gives some approximately correct finite sample bound under hypothesis $\mathbf{MA}(1, C)$. It is easy to extend those results to the more general case of $\mathbf{MA}(\infty, C)$. Note in particular that this assumption is always satisfied for $C = 4$.

Proof of Lemma 2.2 First consider in our case that, the margin assumption is always true for $C = 4$, $\mathbb{E}(q_\theta^2) \leq 4$, the rest of the proof is similar to that of lemma 2.1. From equation (4.3) with the above hypothesis:

$$\rho(R_n(\theta)) - \bar{R}_n \leq \rho(R(\theta)) - \bar{R} + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma} + \frac{4\gamma}{n-1}\frac{1}{(1-\frac{4}{n-1})}$$

From the Bernstein inequality with in the symmetric case we get with probability $1-\epsilon$:

$$\rho(R(\theta)) - \bar{R} \leq \rho(R_n(\theta)) - \bar{R}_n + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma} + \frac{4\gamma}{n-1}\frac{1}{(1-\frac{4}{n-1})}$$

We get, after noting that the Gibbs posterior can be written as an infimum (Legendre transform), with probability $1-2\epsilon$:

$$\int (R(\theta)\pi_\gamma(d\theta|\mathcal{D}) - \bar{R} \leq \inf_{\rho\in\mathcal{M}_+^1(\Theta)}\rho(R(\theta)) - \bar{R} + \frac{\mathcal{K}(\rho,\pi) + \log\frac{1}{\epsilon}}{\gamma} + \frac{16\gamma}{n-1}$$

(we also used $\gamma \leq (n-1)/8$).
$\square$

The two above lemma depend on some class complexity $\mathcal{K}(\rho, \pi)$. The latter can be specialized to different choice of prior measure π. In the following we propose two specifications to a Gaussian prior and a spike and slab prior.

4.A.3 Proof of Theorem 2.3 (Independent Gaussian prior)

For any $\theta_0 \in \mathbb{R}^p$ with $\|\theta_0\| = 1$ and $\delta > 0$ we put

$$\rho_{\theta_0,\delta}(\mathrm{d}\theta) \propto \mathbf{1}_{\|\theta-\theta_0\|\leq\delta}\pi(\mathrm{d}\theta).$$

Then we have, from Lemma 2.1, with probability at least $1-\varepsilon$,

$$\int R(\theta)\pi_\gamma(\mathrm{d}\theta|\mathcal{D}) - \overline{R} \leq 2\inf_{\theta_0,\delta}\left\{\int R(\theta)\rho_{\theta_0,\delta}(\mathrm{d}\theta) - \overline{R} + 16C\frac{\mathcal{K}(\rho_{\theta_0,\delta},\pi) + \log\left(\frac{4}{\varepsilon}\right)}{(n-1)}\right\}$$

First, note that

$$\begin{aligned}
R(\theta) &= \mathbb{E}\left(\mathbb{1}\{\langle\theta, X-X'\rangle\,(Y-Y') < 0\}\right)\\
&= \mathbb{E}\left(\mathbb{1}\{\langle\theta_0, X-X'\rangle\,(Y-Y') < 0\}\right)\\
&\quad + \mathbb{E}\left(\mathbb{1}\{\langle\theta, X-X'\rangle\,(Y-Y') < 0\} - \mathbb{1}\{\langle\theta_0, X-X'\rangle\,(Y-Y') < 0\}\right)\\
&\leq \mathbb{R}(\theta_0) + \mathbb{P}(\operatorname{sign}\langle\theta, X-X'\rangle\,(Y-Y') \neq \operatorname{sign}\langle\theta_0, X-X'\rangle\,(Y-Y'))\\
&= \mathbb{R}(\theta_0) + \mathbb{P}(\operatorname{sign}\langle\theta, X-X'\rangle \neq \operatorname{sign}\langle\theta_0, X-X'\rangle)\\
&\leq R(\theta_0) + c\left\|\frac{\theta}{\|\theta\|} - \theta_0\right\|\\
&\leq R(\theta_0) + 2c\|\theta-\theta_0\|.
\end{aligned}$$

As a consequence $\int R(\theta)\rho_{\theta_0,\delta}(\mathrm{d}\theta) \leq R(\theta_0) + 2c\delta$.

The next step is to calculate $\mathcal{K}(\rho_{\theta_0,\delta},\pi)$. We have

$$\mathcal{K}(\rho_{\theta_0,\delta},\pi) = \log\frac{1}{\pi\left(\{\theta : \|\theta-\theta_0\| \leq \delta\}\right)}.$$

Assuming that $\theta_{0,1} > 0$ (the proof is exactly symmetric in the other case)

$$\begin{aligned}
-\mathcal{K}(\rho_{\theta_0,\delta},\pi) &= \log\pi\left(\{\theta : \sum_{i=1}^{d}(\theta_i-\theta_{0,i})^2 \leq \delta^2\}\right)\\
&\geq d\log\pi\left(\{\theta : (\theta_1-\theta_{0,1})^2 \leq \frac{\delta^2}{d}\}\right)\\
&\geq d\log\int_{\frac{\theta_{0,1}}{\sqrt{\vartheta}}-\frac{\delta}{\sqrt{\vartheta d}}}^{\frac{\theta_{0,1}}{\sqrt{\vartheta}}+\frac{\delta}{\sqrt{\vartheta d}}}\varphi_{(0,1)}(x)\mathrm{d}x\\
&\geq d\log\left(\frac{\delta}{2\sqrt{\vartheta d}}\varphi\left(\frac{\theta_{0,1}}{\sqrt{\vartheta}}+\frac{\delta}{\sqrt{\vartheta d}}\right)\right)\\
&\geq d\log\left(\frac{\delta}{2\sqrt{\vartheta d}}\varphi\left(\frac{1}{\sqrt{\vartheta}}+\frac{\delta}{\sqrt{\vartheta d}}\right)\right)
\end{aligned}$$

$$= d \log \left(\frac{\delta}{2\sqrt{2\pi\vartheta d}} \exp \left[-\frac{1}{2} \left(\frac{1}{\sqrt{\vartheta}} + \frac{\delta}{\sqrt{\vartheta d}} \right)^2 \right] \right)$$
$$\geq d \log \left\{ \frac{\delta}{2\sqrt{2\pi\vartheta d}} \exp \left(-\frac{1}{\vartheta} - \frac{\delta^2}{\vartheta d} \right) \right\}$$

$$\mathcal{K}(\rho_{\theta_0,\delta}, \pi) \leq -d \log\{\delta\} + \frac{d}{2} \log\{8\pi\vartheta d\} + \frac{1}{\vartheta} + \frac{\delta^2}{\vartheta d}$$

And we can plug the equation above in the result of lemma 2.1 with $\delta = \frac{1}{n}$

$$\int R(\theta)\pi_\gamma(\theta|\mathcal{D}) - \bar{R} \leq 2 \inf_{\theta_0} \left(R(\theta_0) - \bar{R} + 2c\frac{1}{n} + \frac{2}{\gamma} \left(d \log\{n\} + \frac{d}{2} \log\{8\pi\vartheta d\} + \frac{1}{\vartheta} + \frac{\frac{1}{n^2}}{\vartheta d} + \log \frac{4}{\epsilon} \right) \right)$$

Any $\gamma = O(n)$ will lead to a convergence result. Taking $\gamma = (n-1)/8C$ and optimizing in ϑ we obtain a variance of $\vartheta = \frac{2(1+\frac{1}{n^2 d})}{d}$.

4.A.4 Proof of Theorem 2.4 (Independent Gaussian prior)

As was done for the previous lemmas we can lift the $\mathbf{MA}(\infty, C)$ and use the lemma 2.2 instead, which gives rise to Theorem 2.4.

Use Lemma 2.2 and the same steps as in the proof of Theorem 4.2.3, optimize w.r.t. γ and ϑ to get the result.

We show the same kind of result in the following but for spike and slab priors.

4.A.5 Proof of Theorem 2.5 (Spike and slab prior for feature selection)

As for the proof of theorem 4.2.3 we start by defining, for any $\theta_0 \in \mathbb{R}^p$ with $\|\theta_0\| = 1$ and $\delta > 0$,

$$\rho_{\theta_0,\delta}(\mathrm{d}\theta) \propto \mathbf{1}_{\|\theta - \theta_0\| \leq \delta} \pi(\mathrm{d}\theta)$$

so that in the end, by a similar argument as previously it remains only to upper bound the following quantity,

$$\mathcal{K}(\rho_{\theta_0,\delta}, \pi) = \log \frac{1}{\pi\left(\{\theta : \|\theta - \theta_0\| \leq \delta\}\right)}.$$

Let π_0 denote the probability distribution such that the θ_i are iid $\mathcal{N}(0, v_0)$. So:

$$-\mathcal{K}(\rho_{\theta_0,\delta}, \pi) = \log \pi \left(\left\{ \theta : \sum_{i=1}^{d} (\theta_i - \theta_{0,i})^2 \leq \delta^2 \right\} \right)$$

$$
\begin{aligned}
&\geq \log \pi \left(\left\{ \theta : \forall i, (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) \\
&= \sum_{i:\theta_{0,i} \neq 0} \log \pi \left(\left\{ (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) \\
&\quad + \log \pi \left(\left\{ \forall i \text{ with } \theta_{0,i} = 0, \theta_i^2 < \frac{\delta^2}{d} \right\} \right) \\
&\geq \sum_{i:\theta_{0,i} \neq 0} \log \pi \left(\left\{ (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) \\
&\quad + \log \pi_0 \left(\left\{ \forall i \text{ with } \theta_{0,i} = 0, \theta_i^2 < \frac{\delta^2}{d} \right\} \right) + d \log(1-p) \\
&= \sum_{i:\theta_{0,i} \neq 0} \log \pi \left(\left\{ (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) \\
&\quad + \log \left[1 - \pi_0 \left(\left\{ \exists i, \theta_{0,i} = 0, \theta_i^2 > \frac{\delta^2}{d} \right\} \right) \right] + d \log(1-p) \\
&\geq \sum_{i:\theta_{0,i} \neq 0} \log \pi \left(\left\{ (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) \\
&\quad + \log \left[1 - \sum_{i:\theta_i = 0} \pi_0 \left(\left\{ \theta_i^2 > \frac{\delta^2}{d} \right\} \right) \right] + d \log(1-p).
\end{aligned}
$$

Assume first that i is such that $\theta_{0,i} = 0$. Then:

$$
\begin{aligned}
\pi_0 \left(\left\{ \theta_i^2 > \frac{\delta^2}{d} \right\} \right) &= \pi_0 \left(\left\{ \left| \frac{\theta_i}{\sqrt{v_0}} \right| > \frac{\delta}{\sqrt{v_0 d}} \right\} \right) \\
&\leq \exp\left(-\frac{\delta^2}{2 v_0 d} \right),
\end{aligned}
$$

and so

$$
\sum_{i:\theta_{0,i}=0} \pi_0 \left(\left\{ \theta_i^2 > \frac{\delta^2}{d} \right\} \right) \leq d \exp\left(-\frac{\delta^2}{2 v_0 d} \right) \leq \frac{1}{2}
$$

as soon as $v_0 \leq \delta^2 / (2d \log(d))$. Then, assume that i is such that $\theta_{0,i} \neq 0$. Now assume that $\theta_{0,i} > 0$ (the proof is exactly symmetric if $\theta_{0,i} < 0$):

$$
\begin{aligned}
\pi \left(\left\{ \theta : (\theta_i - \theta_{0,i})^2 \leq \frac{\delta^2}{d} \right\} \right) &\geq p \int_{\frac{\theta_{0,i}}{\sqrt{v_1}} - \frac{\delta}{\sqrt{v_1 d}}}^{\frac{\theta_{0,i}}{\sqrt{v_1}} + \frac{\delta}{\sqrt{v_1 d}}} \varphi_{(0,1)}(x) \mathrm{d}x \\
&\geq \frac{p\delta}{2\sqrt{v_1 d}} \varphi \left(\frac{\theta_{0,i}}{\sqrt{v_1}} + \frac{\delta}{\sqrt{v_1 d}} \right)
\end{aligned}
$$

$$\geq \frac{p\delta}{2\sqrt{v_1 d}}\varphi\left(\frac{1}{\sqrt{v_1}}+\frac{\delta}{\sqrt{v_1 d}}\right)$$
$$= \frac{p\delta}{2\sqrt{2\pi v_1 d}}\exp\left[-\frac{1}{2}\left(\frac{1}{\sqrt{v_1}}+\frac{\delta}{\sqrt{v_1 d}}\right)^2\right]$$
$$\geq \frac{p\delta}{2\sqrt{2\pi v_1 d}}\exp\left[-\frac{1}{v_1}-\frac{\delta^2}{v_1 d}\right].$$

Putting everything together:

$$\mathcal{K}(\rho_{\theta_0,\delta},\pi) \leq -\|\theta_0\|_0 \log\left(\frac{p\delta}{2\sqrt{2\pi v_1 d}}\exp\left[-\frac{1}{v_1}-\frac{\delta^2}{v_1 d}\right]\right)+\log(2)+d\log\frac{1}{1-p}$$
$$= \|\theta_0\|_0\left[\log\left(\frac{2\sqrt{2\pi v_1 d}}{p\delta}\right)+\frac{1}{v_1}+\frac{\delta^2}{v_1 d}\right]+\log(2)+d\log\frac{1}{1-p}.$$

So, we have:

$$\int R(\theta)\pi_\gamma(\mathrm{d}\theta|\mathcal{D}) - \overline{R} \leq 2\inf_{\theta_0,\delta}\Bigg\{R(\theta_0)-\overline{R}+2c\delta + 16C\frac{\|\theta_0\|_0\left[\log\left(\frac{2\sqrt{2\pi v_1 d}}{p\delta}\right)+\frac{1}{v_1}+\frac{\delta^2}{v_1 d}\right]+\log(2)+d\log\frac{1}{1-p}+\log\left(\frac{4}{\varepsilon}\right)}{(n-1)}\Bigg\}$$

4.B Practical implementation of the PAC-Bayesian approach

4.B.1 Sequential Monte Carlo

The resampling scheme we use in our SMC sampler is systematic resampling, see Algorithm 20.

To move the particles while leaving invariant the current target $\pi_{\xi,\gamma}(\theta|\mathcal{D})$, we use the standard random walk Metropolis strategy, but scaled to the current set of particles, as outlined by Algorithm 17.

4.B.2 Expectation-Propagation (Gaussian prior)

EP aims at approximating posterior distributions of the form,

$$\pi(\theta|\mathcal{D}) = \frac{1}{Z_\pi}P_0(\theta)\prod_{i=1}^{n}t_i(\theta)$$

Algorithm 16 Systematic resampling

Input: Normalised weights $W_t^j := w_t(\theta_{t-1}^j)/\sum_{i=1}^N w_t(\theta_{t-1}^i)$.

Output: indices $A^i \in \{1, \ldots, N\}$, for $i = 1, \ldots, N$.

a. Sample $U \sim \mathcal{U}([0, 1])$.

b. Compute cumulative weights as $C^n = \sum_{m=1}^n NW^m$.

c. Set $s \leftarrow U$, $m \leftarrow 1$.

d. **For** $n = 1 : N$

While $C^m < s$ **do** $m \leftarrow m + 1$.

$A^n \leftarrow m$, and $s \leftarrow s + 1$.

End For

by approximating each site $t_i(\theta)$ by a distribution from an exponential family $q_i(\theta)$. The algorithm cycles through each site, computes the cavity distribution $Q^{\setminus i}(\theta) \propto Q(\theta)q_i^{-1}(\theta)$ and minimizes the Kullback-Leibler divergence between $Q^{\setminus i}(\theta)t_i(\theta)$ and the global approximation $Q(\theta)$. This is efficiently done by using properties of the exponential family (e.g. Bishop [2006a]).

In the Gaussian case the EP approximation can be written as a product of some prior and a product of sites:

$$Q(\theta) \propto \mathcal{N}(\theta; 0, \Sigma) \prod_{i,j} q_{ij}(\theta),$$

for which the sites are unnormalized Gaussians for the natural parametrization $q_{ij}(\theta) \propto \exp\left(-\frac{1}{2}\theta^T Q_{ij}\theta + \theta r_{ij}\right)$. We can equivalently use the one dimensional representation $q_{ij}(s_{ij}) \propto \exp\left(-\frac{1}{2}s_{ij}^2 K_{ij} + s_{ij}h_{ij}\right)$, going from one to the other is easily done by multiplying θ by $(e_i - e_j)X$ where $\forall i \in \{1, \cdots, n\}$, e_i is a vector of zeroes with one on the i-th line. Hence we keep in memory only $(K_{ij})_{ij}$ and $(h_{ij})_{ij}$.

While computing the cavity moment we must compute $(Q - (X_i - X_j)(X_i - X_j)K_{ij})$ and its inverse. The latter can be computed efficiently using Woodbury formula. Equivalently one could use similar tricks where only the Cholesky factorisation is saved and updated as in Seeger [2005a]. By precomputing some matrix multiplication the later cavity moment computation can be done in complexity $\mathcal{O}(p^2)$.

Algorithm 17 Gaussian random walk Metropolis step

Input: θ, S ($d \times d$ positive matrix)

Output: θ_{next}

a. Sample $\theta_{\text{prop}} \sim \mathcal{N}(\theta, S)$.

b. Sample $U \sim \mathcal{U}([0,1])$.

c. If $\log(U) \leq \log \pi_{\xi,\gamma}(\theta_{\text{prop}}|\mathcal{D})/\pi_{\xi,\gamma}(\theta|\mathcal{D})$, set $\theta_{\text{next}} \leftarrow \theta_{\text{prop}}$, otherwise set $\theta_{\text{next}} \leftarrow \theta$.

To update the sites we compute normalizing constant $Z_{ij} = \int \mathcal{N}(s; m^{\setminus ij}, \sigma^{\setminus ij}) t_{ij}(s) \mathrm{d}s$ and use properties of exponential families.

Normalising Constant The normalizing constant of the posterior can be computed using EP. We have that for each sites $t_{ij}(\theta) = C_{ij} q_{ij}(\theta)$ we replace those sites in integral we wish to approximate,

$$\int \mathcal{N}(\theta; 0, \Sigma) \prod_{ij} t_{ij}(\theta) \mathrm{d}\theta \simeq \prod_{ij} C_{ij} \int \mathcal{N}(\theta; 0, \Sigma) \prod_{ij} q_{ij}(\theta) \mathrm{d}\theta$$

The integral on the right hand side is a Gaussian convolution and is therefore also Gaussian. The C_{ij}s can be approximated by matching the zeroth order moment in the site update. As noted in the chapter we can also compute the derivatives with respect to some prior hyper-parameter (see Seeger [2005a]).

4.B.3 Expectation-Propagation (spike and slab prior)

The posterior can be written as

$$\pi(\theta|\mathcal{D}) \propto \prod_{i,j} t_{ij}(\theta) \prod_{k=1}^{d} t_k(\theta_k, z_k) \mathcal{B}er(z_k; p),$$

where $z_k \in \{0,1\}$ codes the origin of θ_k, spike/slab, and where $t_k(\theta_k, z_k) \propto z_k \mathcal{N}(\theta_k; 0, v_0) + (1 - z_k)\mathcal{N}(\theta_k; 0, v_1)$. The approximation given by EP is of the form,

$$Q(\theta, z) \propto \prod_{i,j} q_{ij}(\theta) \prod_{k=1}^{d} q_k(\theta_k, z_k) \mathcal{B}er(z_k; p_k),$$

Algorithm 18 parallel EP for Gaussian Prior

Input: ϑ, γ

Output: m and V

Init: $V \leftarrow \Sigma$, $m \leftarrow 0$

Untill Convergence **Do**

For all sites (i, j)**Do** in parallel

a. Compute the cavity moments $m^{\backslash ij}$, $V^{\backslash ij}$

b. Compute the 1st and 2nd order moments of $q^{\backslash ij}(s_{ij})t_{ij}(s_ij)$

c. Update K_{ij} and h_{ij}

End For

Update $V = (\Sigma^{-1} + \sum_{ij}(X_i - X_j)^T(X_i - X_j)K_{ij})^{-1}$, $m = V(\sum_{ij}(X_i - X_j)h_{ij})$

End While

where $q_k(\theta_k, z_k) \propto \mathcal{B}er(z_k, p_k)\mathcal{N}(\theta_k; m_k, \sigma_k^2)$, and $t_{ij}(\theta)$ is as in the previous section. The cavity moments are easy to compute as the approximation is Gaussian in θ and Bernoulli in z. In both cases we can deduce cavity moments because division is stable inside those classes of functions. We get some distribution $Q^{\backslash k}(\theta_k) \propto \mathcal{B}er(z_k; p^{\backslash k})\mathcal{N}(\theta_k; m^{\backslash k}, \sigma^{2,\backslash k})$. We can compute the normalizing constant of the distribution $Q^{\backslash ij}(\theta)t_k(\theta_k, z_k)$, namely,

$$Z_k = p^{\backslash k} \int \mathcal{N}(\theta_k; 0, v_0)\mathcal{N}(\theta_k; m^{\backslash k}, \sigma^{2,\backslash k})\mathrm{d}\theta_k + (1-p^{\backslash k}) \int \mathcal{N}(\theta_k; 0, v_0)\mathcal{N}(\theta_k; m^{\backslash k}, \sigma^{2,\backslash k})\mathrm{d}\theta_k$$

Where we can find the update by computing the derivatives of $\log Z_k$ with respect to $p^{\backslash k}$, $m^{\backslash k}$ and $\sigma^{2,\backslash k}$

Initialization for the Gaussian is done to a given Σ_0 that will be subtracted later on. The initial p_ks are taken such that the approximation equals the prior p at the first iteration.

4.C Numerical illustration

Figure 4.C.1 shows the posterior marginals as given by EP and tempering SMC. The later is exact in the sense that the only error stems from Monte Carlo; we

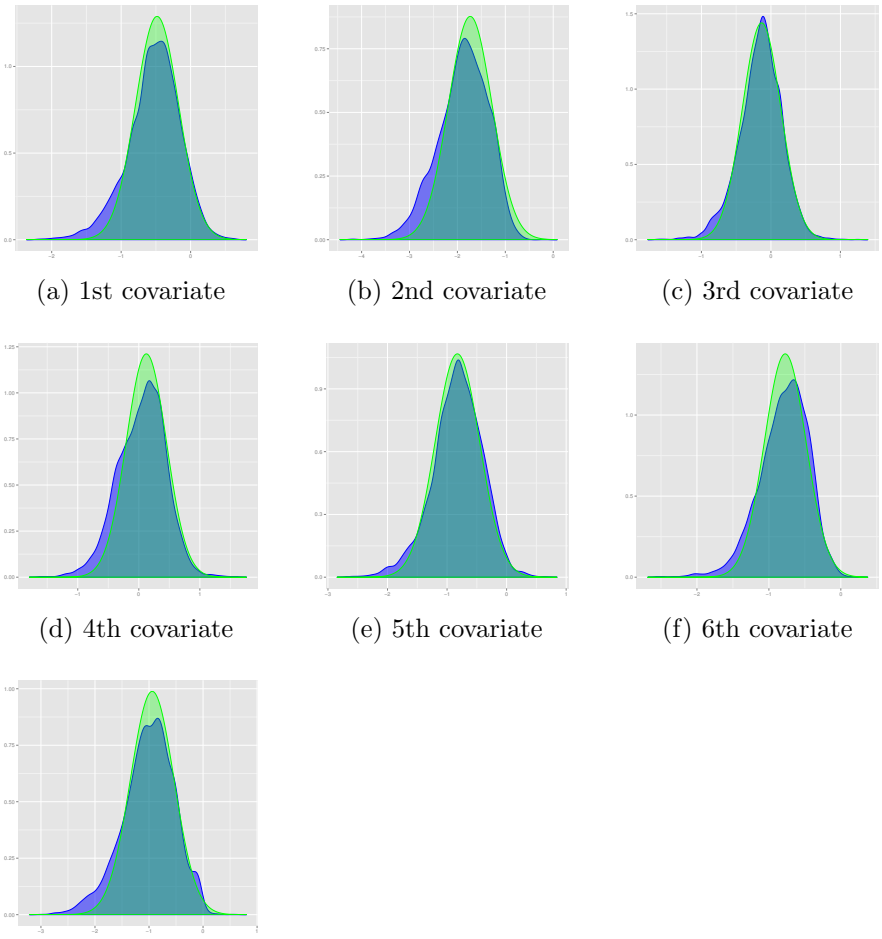
see that the mode is well approximated however the variance is slightly underestimated.

In Table 4.C.1 we show the CPU times in seconds, on all dataset studied. Experiments where run with a i7-3720QM CPU @ 2.60GHz intel processor with 6144 KB cache. Our linear model is overall faster on those datasets. A caveat is that Rankboost is implemented in Matlab, while our implementation is in C.

Dataset	**Covariates**	**Balance**	**EP-AUC**	**GPEP-AUC**	**Rankboost**
Pima	7	34%	**0.06**	7.75	3.26
Credit	60	28%	**1.98**	7.59	56.54
DNA	180	22%	**11.26**	63.47	141.60
SPECTF	22	50%	**0.25**	63.47	3.55
Colon	2000	40%	636.63	**60.99**	156.85
Glass	10	1%	**0.23**	1.33	2.36

Table 4.C.1: Computation times in seconds

Figure 4.C.1: Comparison of the output of the two algorithms



(a) 1st covariate (b) 2nd covariate (c) 3rd covariate

(d) 4th covariate (e) 5th covariate (f) 6th covariate

(g) 7th covariate

Comparison of the Gaussian approximation obtained by Fractional EP (green) with the true density generated by SMC (blue) on the Pima indians dataset

5

Properties of variational approximations of Gibbs posteriors

This is joint work with Pierre Alquier and Nicolas Chopin
Status: Submitted to *Journal of Machine Learning Rechearch*

5.1 Introduction

A Gibbs posterior, also known as a PAC-Bayesian or pseudo-posterior, is a probability distribution for random estimators of the form:

$$\hat{\rho}_\lambda(\mathrm{d}\theta) = \frac{\exp[-\lambda r_n(\theta)]}{\int \exp[-\lambda r_n]\mathrm{d}\pi}\pi(\mathrm{d}\theta).$$

More precise definitions will follow, but for now, θ may be interpreted as a parameter (in a finite or infinite-dimensional space), $r_n(\theta)$ as an empirical measure of risk (e.g. prediction error), and $\pi(\mathrm{d}\theta)$ a prior distribution.

We will follow in this chapter the PAC (Probably Approximatively Correct)-Bayesian approach, which originates from machine learning [Catoni, 2004; McAllester, 1998; Shawe-Taylor and Williamson, 1997]; see Catoni [2007] for an exhaustive study, and Dalalyan and Tsybakov [2008]; Jiang and Tanner [2008]; Yang [2004]; Zhang [2006] for related perspectives (such as the aggregation of estimators in the last 3 papers). There, $\hat{\rho}_\lambda$ appears as the probability distribution that minimises the upper bound of an oracle inequality on the risk of *random* estimators. The PAC-Bayesian approach offers sharp theoretical guarantees on the properties of such estimators, without assuming a particular model for the data generating process.

The Gibbs posterior has also appeared in other places, and under different motivations: in Econometrics, as a way to avoid direct maximisation in moment estimation [Chernozhukov and Hong, 2003]; and in Bayesian decision theory, as as way to define a Bayesian posterior distribution when no likelihood has been specified [Bissiri et al., 2013]. Another well-known connection, although less directly useful (for Statistics), is with thermodynamics, where r_n is interpreted as an energy function, and λ as the inverse of a temperature.

Whatever the perspective, estimators derived from Gibbs posteriors usually show excellent performance in diverse tasks, such as classification, regression, ranking, and so on, yet their actual implementation is still far from routine. The usual recommendation [Alquier and Biau, 2013; Dalalyan and Tsybakov, 2012; Guedj and Alquier, 2013] is to sample from a Gibbs posterior using MCMC [Markov chain Monte Carlo, see e.g. Green et al., 2015]; but constructing an efficient MCMC sampler is often difficult, and even efficient implementations are often too slow for practical uses when the dataset is very large.

In this chapter, we consider instead VB (Variational Bayes) approximations, which have been initially developed to provide fast approximations of 'true' posterior distributions (i.e. Bayesian posterior distributions for a given model); see Jordan et al. [1999]; MacKay [2002] and Chap. 10 in Bishop [2006a].

Our main results are as follows: when PAC-Bayes bounds are available - mainly, when a strong concentration inequality holds - replacing the Gibbs posterior by a variational approximation does not affect the rate of convergence to the best possible prediction, on the condition that the Küllback-Leibler divergence between the posterior and the approximation is itself controlled in an appropriate way.

We also provide empirical bounds, which may be computed from the data so as to ascertain the actual performance of estimators obtained by variational approximation. All the results gives strong incentives, we believe, to recommend Variational Bayes as the default approach to approximate Gibbs posteriors.

The rest of the chapter is organized as follows. In Section 5.2 we introduce the notations and assumptions. In Section 5.3 we introduce variational approximations and the corresponding algorithms. The main results are provided in general form in Section 5.4: in Subsection 5.4.1, we give results under the assumption that a Hoeffding type inequality holds (slow rates) and in Subsection 5.4.2, we give results under the assumption that a Bernstein type inequality holds (fast rates). Note that for the sake of shortness, we will refer to these settings as "Hoeffding assumption" and "Bernstein assumption" even if this terminology is non standard. We then apply these results in various settings: classification (Section 5.5), convex classification (Section 5.6), ranking (Section 5.7), and matrix completion (Section 5.8). In each case, we show how to specialise the general results of Section 5.4 to the considered application, so as to obtain the properties of the VB

approximation, and we also discuss its numerical implementation. All the proofs are collected in the Appendix.

5.2 PAC-Bayesian framework

We observe a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$, taking values in $\mathcal{X} \times \mathcal{Y}$, where the pairs (X_i, Y_i) have the same distribution P. We will assume explicitly that the (X_i, Y_i)'s are independent in several of our specialised results, but we do not make this assumption at this stage, as some of our general results, and more generally the PAC-Bayesian theory, may be extended to dependent observations; see e.g. Alquier and Li [2012]. The label set $\mathcal{Y}$ is always a subset of $\mathbb{R}$. A set of predictors is chosen by the statistician: $\{f_\theta : \mathcal{X} \to \mathbb{R},\, \theta \in \Theta\}$. For example, in linear regression, we may have: $f_\theta(x) = \langle \theta, x \rangle$, the inner product of $\mathcal{X} = \mathbb{R}^d$, while in classification, one may have $f_\theta(x) = \mathbb{I}_{\langle \theta, x \rangle > 0} \in \{0, 1\}$.

We assume we have at our disposal a risk function $R(\theta)$; typically $R(\theta)$ is a measure of the prevision error. We set $\overline{R} = R(\overline{\theta})$, where $\overline{\theta} \in \arg\min_\Theta R$; i.e. $f_{\overline{\theta}}$ is an optimal predictor. We also assume that the risk function $R(\theta)$ has an empirical counterpart $r_n(\theta)$, and set $\overline{r}_n = r_n(\overline{\theta})$. Often, R and r_n are based on a loss function $\ell : \mathbb{R}^2 \to \mathbb{R}$; i.e. $R(\theta) = \mathbb{E}[\ell(Y, f_\theta(X))]$ and $\overline{r}_n(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f_\theta(X_i))$. (In this chapter, the symbol $\mathbb{E}$ will always denote the expectation with respect to the (unknown) law P of the (X_i, Y_i)'s.) There are situations however (e.g. ranking), where R and r_n have a different form.

We define a prior probability measure $\pi(\cdot)$ on the set Θ (equipped with the standard σ-algebra for the considered context), and we let $\mathcal{M}_+^1(\Theta)$ denote the set of all probability measures on Θ.

Definition 5.2.1 *We define, for any $\lambda > 0$, the pseudo-posterior $\hat{\rho}_\lambda$ by*

$$\hat{\rho}_\lambda(\mathrm{d}\theta) = \frac{\exp[-\lambda r_n(\theta)]}{\int \exp[-\lambda r_n]\mathrm{d}\pi}\pi(\mathrm{d}\theta).$$

The pseudo-posterior $\hat{\rho}_\lambda$ (also known as the Gibbs posterior, Catoni [2004, 2007], or the exponentially weighted aggregate, Dalalyan and Tsybakov [2008]) plays a central role in the PAC-Bayesian approach. It is obtained as the distribution that minimises the upper bound of a certain oracle inequality applied to *random* estimators. Practical estimators (predictors) may be derived from the pseudo-posterior, by e.g. taking the expectation, or sampling from it. Of course, when $\exp[-\lambda r_n(\theta)]$ may be interpreted as the likelihood of a certain model, $\hat{\rho}_\lambda$ becomes a Bayesian posterior distribution, but we will not restrict our attention to this particular case.

The following 'theoretical' counterpart of $\hat{\rho}_\lambda$ will prove useful to state results.

Definition 5.2.2 *We define, for any $\lambda > 0$, π_λ as*

$$\pi_\lambda(\mathrm{d}\theta) = \frac{\exp[-\lambda R(\theta)]}{\int \exp[-\lambda R]\mathrm{d}\pi}\pi(\mathrm{d}\theta).$$

We will derive PAC-Bayesian bounds on predictions obtained by variational approximations of $\hat{\rho}_\lambda$ under two types of assumptions: a Hoeffding-type assumption, from which we may deduce slow rates of convergence (Subsection 5.4.1), and a Bernstein-type assumption, from which we may obtain fast rates of convergence (Subsection 5.4.2).

Definition 5.2.3 *We say that a Hoeffding assumption is satisfied for prior π when there is a function f and an interval $I \subset \mathbb{R}_+^*$ such that, for any $\lambda \in I$,*

$$\left.\begin{array}{l} \pi\left(\mathbb{E}\exp\left\{\lambda[R(\theta) - r_n(\theta)]\right\}\right) \\ \pi\left(\mathbb{E}\exp\left\{\lambda[r_n(\theta) - R(\theta)]\right\}\right) \end{array}\right\} \leq \exp\left[f(\lambda, n)\right]. \tag{5.1}$$

Inequality (5.1) can be interpreted as an integrated version (with respect to π) of Hoeffding's inequality, for which $f(\lambda, n) \asymp \lambda^2/n$. In many cases the loss will be bounded uniformly over θ; then Hoeffding's inequality will directly imply (5.1). The expectation with respect to π in (5.1) allows us to treat some cases where the loss is not upper bounded by specifying a prior with sufficiently light tails.

Definition 5.2.4 *We say that a Bernstein assumption is satisfied for prior π when there is a function g and an interval $I \subset \mathbb{R}_+^*$ such that, for any $\lambda \in I$,*

$$\left.\begin{array}{l} \pi\left(\mathbb{E}\exp\left\{\lambda[R(\theta) - \overline{R}] - \lambda[r_n(\theta) - \overline{r}_n]\right\}\right) \\ \pi\left(\mathbb{E}\exp\left\{\lambda[r_n(\theta) - \overline{r}_n] - \lambda[R(\theta) - \overline{R}]\right\}\right) \end{array}\right\} \leq \pi\left(\exp\left[g(\lambda, n)[R(\theta) - \overline{R}]\right]\right). \tag{5.2}$$

This assumption is satisfied for example by sums of i.i.d. sub-exponential random variables, see Subsection 2.4 p. 27 in Boucheron et al. [2013], when a margin assumption on the function $R(\cdot)$ is satisfied [Tsybakov, 2004]. This is discussed in Section 5.4.2. Again, extensions beyond the i.i.d. case are possible, see e.g. Wintenberger [2010] for a survey and new results. In all these examples, the important feature of the function g that we will use to derive rates of convergence is the fact that there is a constant $c > 0$ such that when $\lambda = cn$, $g(\lambda, n) = g(cn, n) \asymp n$.

As mentioned previously, we will often consider $r_n(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f_\theta(X_i))$, however, the previous assumptions can also be satisfied when $r_n(\theta)$ is a U-statistic, using Hoeffding's decomposition of U-statistics combined with the corresponding inequality for sums of independent variables [Hoeffding, 1948]. This idea comes from Clémençon et al. [2008a] and we will use it in our ranking application.

Remark 5.2.1 *We could consider more generally inequalities of the form*

$$\left.\begin{array}{l}\pi\left(\mathbb{E}\exp\left\{\lambda[R(\theta)-\overline{R}]-\lambda[r_n(\theta)-\overline{r}_n]\right\}\right)\\ \pi\left(\mathbb{E}\exp\left\{\lambda[r_n(\theta)-\overline{r}_n]-\lambda[R(\theta)-\overline{R}]\right\}\right)\end{array}\right\}\leq\pi\left(\exp\left[g(\lambda,n)[R(\theta)-\overline{R}]^{\kappa}\right]\right)$$

that allow to use the more general form of the margin assumption of Mammen and Tsybakov [1999]; Tsybakov [2004]. PAC-Bayes bounds in this context are provided by Catoni [2007]. However, the techniques involved would require many pages to be described so we decided to focus on the cases $\kappa = 0$ and $\kappa = 1$ to keep the exposition simple.

5.3 Numerical approximations of the pseudo-posterior

5.3.1 Monte Carlo

As already explained in the introduction, the usual approach to approximate $\hat{\rho}_\lambda$ is MCMC (Markov chain Monte Carlo) sampling. Ridgway [2015] proposed tempering SMC (Sequential Monte Carlo, e.g. Del Moral et al. [2006b]) as an alternative to MCMC to sample from Gibbs posteriors: one samples sequentially from $\hat{\rho}_{\lambda_t}$, with $0 = \lambda_0 < \cdots < \lambda_T = \lambda$ where λ is the desired temperature. One advantage of this approach is that it makes it possible to contemplate different values of λ, and choose one by e.g. cross-validation. Another advantage is that such an algorithm requires little tuning; see Appendix B for more details on the implementation of tempering SMC. We will use tempering SMC as our gold standard in our numerical studies.

SMC and related Monte Carlo algorithms tend to be too slow for practical use in situations where the sample size is large, the dimension of Θ is large, or f_θ is expensive to compute. This motivates the use of fast, deterministic approximations, such as Variational Bayes, which we describe in the next section.

5.3.2 Variational Bayes

Various versions of VB (Variational Bayes) have appeared in the literature, but the main idea is as follows. We define a family $\mathcal{F} \subset \mathcal{M}_+^1(\Theta)$ of probability distributions that are considered as tractable. Then, we define the VB-approximation of $\hat{\rho}_\lambda$: $\tilde{\rho}_\lambda$.

Definition 5.3.1 *Let*

$$\tilde{\rho}_\lambda = \arg\min_{\rho\in\mathcal{F}} \mathcal{K}(\rho, \hat{\rho}_\lambda),$$

where $\mathcal{K}(\rho, \hat{\rho}_\lambda)$ denotes the KL (Küllback-Leibler) divergence of $\hat{\rho}_\lambda$ relative to ρ: $\mathcal{K}(m,\mu) = \int \log[\frac{\mathrm{d}m}{\mathrm{d}\mu}]\mathrm{d}m$ if $m \ll \mu$ (i.e. μ dominates m), $\mathcal{K}(m,\mu) = +\infty$ otherwise.

The difficulty is to find a family $\mathcal{F}$ (a) which is large enough, so that $\tilde{\rho}_\lambda$ may be close to $\hat{\rho}_\lambda$, and (b) such that computing $\tilde{\rho}_\lambda$ is feasible. We now review two types of families popular in the VB literature.

- Mean field VB: for a certain decomposition $\Theta = \Theta_1 \times \cdots \times \Theta_d$, $\mathcal{F}$ is the set of product probability measures

$$\mathcal{F}^{\mathrm{MF}} = \left\{ \rho \in \mathcal{M}_+^1(\Theta) : \rho(\mathrm{d}\theta) = \prod_{i=1}^{d} \rho_i(\mathrm{d}\theta_i), \forall i \in \{1, \ldots, d\}, \rho_i \in \mathcal{M}_+^1(\Theta_i) \right\}. \tag{5.3}$$

  The infimum of the KL divergence $\mathcal{K}(\rho, \hat{\rho}_\lambda)$, relative to $\rho = \prod_i \rho_i$ satisfies the following fixed point condition [Bishop, 2006a; Parisi, 1988, Chap. 10]:

$$\forall j \in \{1, \cdots, d\} \quad \rho_j(\mathrm{d}\theta_j) \propto \exp\left(\int \{-\lambda r_n(\theta) + \log \pi(\theta)\} \prod_{i \neq j} \rho_i(\mathrm{d}\theta_i) \right) \pi(\mathrm{d}\theta_j). \tag{5.4}$$

  This leads to a natural algorithm were we update successively every ρ_j until stabilization.

- Parametric family:

$$\mathcal{F}^{\mathrm{P}} = \left\{ \rho \in \mathcal{M}_+^1(\Theta) : \rho(\mathrm{d}\theta) = f(\theta; m)\mathrm{d}\theta, m \in M \right\};$$

  and M is finite-dimensional; say $\mathcal{F}^{\mathrm{P}}$ is the family of Gaussian distributions (of dimension d). In this case, several methods may be used to compute the infimum. As above, one may used fixed-point iteration, provided an equation similar to (5.4) is available. Alternatively, one may directly maximize $\int \log[\exp[-\lambda r_n(\theta)]\frac{\mathrm{d}\pi}{\mathrm{d}\rho}(\theta)]\rho(\mathrm{d}\theta)$ with respect to paramater m, using numerical optimization routines. This approach was used for instance in Hoffman et al. [2013] with combination of some stochastic gradient descent to perform inference on a latent Dirichlet allocation model. See also e.g. Emtiyaz Khan et al. [2013]; Khan [2014] for efficient algorithms for Gaussian variational approximation.

In what follows (Subsections 5.4.1 and 5.4.2) we provide tight bounds for the prevision risk of $\tilde{\rho}_\lambda$. This leads to the identification of a condition on $\mathcal{F}$ such that the risk of $\tilde{\rho}_\lambda$ is not worse than the risk of $\hat{\rho}_\lambda$. We will make this condition explicit in various examples, using either mean field VB or parametric approximations.

Remark 5.3.1 *An useful identity, obtained by direct calculations, is: for any* $\rho \ll \pi$,

$$\log \int \exp\left[-\lambda r_n(\theta)\right] \pi(\mathrm{d}\theta) = -\lambda \int r_n(\theta)\rho(\mathrm{d}\theta) - \mathcal{K}(\rho, \pi) + \mathcal{K}(\rho, \hat{\rho}_\lambda). \tag{5.5}$$

Since the left hand side does not depend on ρ, one sees that $\tilde{\rho}_\lambda$, which minimises $\mathcal{K}(\rho, \hat{\rho}_\lambda)$ over $\mathcal{F}$, is also the minimiser of:

$$\tilde{\rho}_\lambda = \arg\min_{\rho \in \mathcal{F}} \left\{ \int r_n(\theta)\rho(\mathrm{d}\theta) + \frac{1}{\lambda}\mathcal{K}(\rho, \pi) \right\}$$

This equation will appear frequently in the sequel in the form of an empirical upper bound.

5.4 General results

This section gives our general results, under either a Hoeffding Assumption (Definition 5.2.3) or a Bernstein Assumption (Definition 5.2.4), on risks bounds for the variational approximation, and how it relates to risks bounds for Gibbs posteriors. These results will be specialised to several learning problems in the following sections.

5.4.1 Bounds under the Hoeffding assumption

Empirical bounds

Theorem 5.4.1 *Under the Hoeffding assumption (Definition 5.2.3), for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously for any $\rho \in \mathcal{M}_+^1(\Theta)$,*

$$\int R\mathrm{d}\rho \leq \int r_n \mathrm{d}\rho + \frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}.$$

This result is a simple variant of a result in Catoni [2007] but for the sake of completeness, its proof is given in Appendix 5.A. It gives us an upper bound on the risk of both the pseudo-posterior (take $\rho = \hat{\rho}_\lambda$) and its variational approximation (take $\rho = \tilde{\rho}_\lambda$). These bounds may be be computed from the data, and therefore provide a simple way to evaluate the performance of the corresponding procedure, in the spirit of the first PAC-Bayesian inequalities [McAllester, 1999, 1998; Shawe-Taylor and Williamson, 1997]. However, this bound do not provide the rate of convergence of these estimators. For this reason, we also provide oracle-type inequalities.

Oracle-type inequalities

Another way to use PAC-Bayesian bounds is to compare $\int R\mathrm{d}\hat{\rho}_\lambda$ to the best possible risk, thus linking this approach to oracle inequalities. This is the point of view developed in Catoni [2004, 2007]; Dalalyan and Tsybakov [2008].

Theorem 5.4.2 *Assume that the Hoeffding assumption is satisfied (Definition 5.2.3). For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously*

$$\int R\mathrm{d}\hat{\rho}_\lambda \leq \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) := \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \int R\mathrm{d}\rho + 2\frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}$$

and

$$\int R\mathrm{d}\tilde{\rho}_\lambda \leq \mathcal{B}_\lambda(\mathcal{F}) := \inf_{\rho \in \mathcal{F}} \left\{ \int R\mathrm{d}\rho + 2\frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}.$$

Moreover,

$$\mathcal{B}_\lambda(\mathcal{F}) = \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}})$$

where we remind that π_λ is defined in Definition 5.2.2.

In this way, we are able to compare $\int R\mathrm{d}\hat{\rho}_\lambda$ to the best possible aggregation procedure in $\mathcal{M}_+^1(\Theta)$ and $\int R\mathrm{d}\tilde{\rho}_\lambda$ to the best aggregation procedure in $\mathcal{F}$. More importantly, we are able to obtain explicit expressions for the right-hand side of these inequalities in various models, and thus to obtain rates of convergence. This will be done in the remaining sections. This leads to the second interest of this result: if there is a $\lambda = \lambda(n)$ that leads to $\mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) \leq \overline{R} + s_n$ with $s_n \to 0$ for the pseudo-posterior $\hat{\rho}_\lambda$, then we only have to prove that there is a $\rho \in \mathcal{F}$ such that $\mathcal{K}(\rho, \pi_\lambda)/\lambda \leq cs_n$ for some constant $c > 0$ to ensure that the VB approximation $\tilde{\rho}_\lambda$ also reaches the rate s_n.

We will see in the following sections several examples where the approximation does not deteriorate the rate of convergence. But first let us show the equivalent oracle inequality under the Bernstein assumption.

5.4.2 Bounds under the Bernstein assumption

In this context the empirical bound on the risk would depend on the minimal achievable risk $\bar{r}_n$, and cannot be computed explicitly. We give the oracle inequality for both the Gibbs posterior and its VB approximation in the following theorem.

Theorem 5.4.3 *Assume that the Bernstein assumption is satisfied (Definition 5.2.4). Assume that $\lambda > 0$ satisfies $\lambda - g(\lambda, n) > 0$. Then for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously:*

$$\int R\mathrm{d}\hat{\rho}_\lambda - \overline{R} \leq \overline{\mathcal{B}}_\lambda\left(\mathcal{M}_+^1(\Theta)\right),$$

$$\int R\mathrm{d}\tilde{\rho}_\lambda - \overline{R} \leq \overline{\mathcal{B}}_\lambda(\mathcal{F}),$$

where, for either $\mathcal{A} = \mathcal{M}_+^1(\Theta)$ *or* $\mathcal{A} = \mathcal{F}$,

$$\overline{\mathcal{B}}_\lambda(\mathcal{A}) = \frac{1}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{A}} \left\{ [\lambda + g(\lambda, n)] \int (R - \overline{R}) \mathrm{d}\rho + 2\mathcal{K}(\rho, \pi) + 2\log\left(\frac{2}{\varepsilon}\right) \right\}.$$

In addition,

$$\overline{\mathcal{B}}_\lambda(\mathcal{F}) = \overline{\mathcal{B}}_\lambda\left(\mathcal{M}_+^1(\Theta)\right) + \frac{2}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{F}} \mathcal{K}\left(\rho, \pi_{\frac{\lambda + g(\lambda,n)}{2}}\right).$$

The main difference with Theorem 5.4.2 is that the function $R(\cdot)$ is replaced by $R(\cdot) - \overline{R}$. This is well known way to obtain better rates of convergence.

5.5 Application to classification

5.5.1 Preliminaries

In all this section, we assume that $\mathcal{Y} = \{0, 1\}$ and we consider linear classification: $\Theta = \mathcal{X} = \mathbb{R}^d$, $f_\theta(x) = \mathbf{1}_{\langle \theta, x \rangle \geq 0}$. We put $r_n(\theta) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{\{f_\theta(X_i) \neq Y_i\}}$, $R(\theta) = \mathbb{P}(Y \neq f_\theta(X))$ and assume that the $[(X_i, Y_i)]_{i=1}^n$ are i.i.d. In this setting, it is well-known that the Hoeffding assumption always holds. We state as a reminder the following lemma.

Lemma 5.5.1 *Hoeffding assumption* (5.1) *is satisfied with* $f(\lambda, n) = \lambda^2/(2n)$.

The proof is given in Appendix 5.A for the sake of completeness.

It is also possible to prove that Bernstein assumption (5.2) holds in the case where the so-called margin assumption of Mammen and Tsybakov is satisfied. This condition we use was introduced by Tsybakov [2004] in a classification setting, based on a related definition in Mammen and Tsybakov [1999].

Lemma 5.5.2 *Assume that Mammen and Tsybakov's margin assumption is satisfied: i.e. there is a constant* C *such that*

$$\mathbb{E}[(\mathbf{1}_{f_\theta(X) \neq Y} - \mathbf{1}_{f_{\overline{\theta}}(X) \neq Y})^2] \leq C[R(\theta) - \overline{R}].$$

Then Bernstein assumption (5.2) *is satisfied with* $g(\lambda, n) = \frac{C\lambda^2}{2n - \lambda}$.

Remark 5.5.1 *We refer the reader to Tsybakov [2004] for a proof that*

$$\mathbb{P}(0 < |\left\langle \overline{\theta}, X \right\rangle| \leq t) \leq C't$$

for some constant $C' > 0$ implies the margin assumption. In words, when X is not likely to be in the region $\left\langle \overline{\theta}, X \right\rangle \simeq 0$, where points are hard to classify, then the problem becomes easier and the classification rate can be improved.

We propose in this context a Gaussian prior: $\pi = \mathcal{N}_d(0, \vartheta^2 I_d)$, and we consider a VB approach based on Gaussian families. The corresponding optimization problem is not convex, but remains feasible as we explain below.

5.5.2 Three sets of Variational Gaussian approximations

Consider the three following Gaussian families

$$\begin{aligned}
\mathcal{F}_1 &= \left\{ \Phi_{\mathbf{m},\sigma^2},\ \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in \mathbb{R}_+^* \right\}, \\
\mathcal{F}_2 &= \left\{ \Phi_{\mathbf{m},\boldsymbol{\sigma}^2},\ \mathbf{m} \in \mathbb{R}^d, \boldsymbol{\sigma}^2 \in (\mathbb{R}_+^*)^2 \right\} \text{ (mean field approximation)}, \\
\mathcal{F}_3 &= \left\{ \Phi_{\mathbf{m},\Sigma},\ \mathbf{m} \in \mathbb{R}^d, \Sigma \in \mathcal{S}^{d+} \right\} \text{ (full covariance approximation)},
\end{aligned}$$

where $\Phi_{\mathbf{m},\sigma^2}$ is Gaussian distribution $N_d(\mathbf{m}, \sigma^2 I_d)$, $\Phi_{\mathbf{m},\boldsymbol{\sigma}^2}$ is $N_d(\mathbf{m}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, and $\Phi_{\mathbf{m},\Sigma}$ is $N_d(\mathbf{m}, \boldsymbol{\Sigma})$. Obviously, $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \mathcal{M}_+^1(\Theta)$, and

$$\mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) \leq \mathcal{B}_\lambda(\mathcal{F}_3) \leq \mathcal{B}_\lambda(\mathcal{F}_2) \leq \mathcal{B}_\lambda(\mathcal{F}_1). \tag{5.6}$$

Note that, for the sake of simplicity, we will use the following classical notations in the rest of the chapter: $\varphi(\cdot)$ is the density of $\mathcal{N}(0,1)$ w.r.t. the Lebesgue measure, and $\Phi(\cdot)$ the corresponding c.d.f. The rest of Section 5.5 is organized as follows. In Subsection 5.5.3, we calculate explicitly $\mathcal{B}_\lambda(\mathcal{F}_2)$ and $\mathcal{B}_\lambda(\mathcal{F}_1)$. Thanks to (5.6) this also gives an upper bound on $\mathcal{B}_\lambda(\mathcal{F}_3)$ and proves the validity of the three types of Gaussian approximations. Then, we give details on algorithms to compute the variational approximation based on $\mathcal{F}_2$ and $\mathcal{F}_3$, and provide a numerical illustration on real data.

5.5.3 Theoretical analysis

We start with the empirical bound for $\mathcal{F}_2$ (and $\mathcal{F}_1$ as a consequence), which is a direct corollary of Theorem 5.4.1.

Corollary 5.5.3 *For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have, for any $\boldsymbol{m} \in \mathbb{R}^d$, $\boldsymbol{\sigma}^2 \in (\mathbb{R}_+)^d$,*

$$\int R \mathrm{d}\Phi_{\mathbf{m},\boldsymbol{\sigma}^2} \leq \int r_n \mathrm{d}\Phi_{\mathbf{m},\boldsymbol{\sigma}^2} + \frac{\lambda}{2n} + \frac{\sum_{i=1}^d \left[\frac{1}{2} \log\left(\frac{\vartheta^2}{\sigma_i^2} \right) + \frac{\sigma_i^2}{\vartheta^2} \right] + \frac{\|\boldsymbol{m}\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}.$$

We now want to apply Theorem 5.4.2 in this context. In order to do so, we introduce an additional assumption.

Definition 5.5.1 *We say that Assumption A1 is satisfied when there is a constant $c > 0$ such that, for any $(\theta, \theta') \in \Theta^2$ with $\|\theta\| = \|\theta'\| = 1$, $\mathbb{P}(\langle X, \theta\rangle \langle X, \theta'\rangle < 0) \leq c\|\theta - \theta'\|$.*

Note that this is not a stringent assumption. For example, it is satisfied as soon as $X/\|X\|$ has a bounded density on the unit sphere.

Corollary 5.5.4 *Assume that the VB approximation is done on either $\mathcal{F}_1$, $\mathcal{F}_2$ or $\mathcal{F}_3$. Take $\lambda = \sqrt{nd}$ and $\vartheta = \frac{1}{\sqrt{d}}$. Under Assumption A1, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously*

$$\left.\begin{array}{l} \int R\mathrm{d}\hat{\rho}_\lambda \\ \int R\mathrm{d}\tilde{\rho}_\lambda \end{array}\right\} \leq \overline{R} + \sqrt{\frac{d}{n}\log\left(4ne^2\right)} + \frac{c}{\sqrt{n}} + \sqrt{\frac{d}{4n^3}} + \frac{2\log\left(\frac{2}{\varepsilon}\right)}{\sqrt{nd}}.$$

See the appendix for a proof. Note also that the values $\lambda = \sqrt{nd}$ and $\vartheta = \frac{1}{\sqrt{d}}$ allow to derive this almost optimal rate of convergence, but are not necessarily the best choices in practice.

Remark 5.5.2 *Note that Assumption A1 is not necessary to obtain oracle inequalities on the risk integrated under $\hat{\rho}_\lambda$. We refer the reader to Chapter 1 in Catoni [2007] for such assumption-free bounds. However, it is clear that without this assumption the shape of $\hat{\rho}_\lambda$ and $\tilde{\rho}_\lambda$ might be very different. Thus, it seems reasonable to require that A1 is satisfied for the approximation of $\hat{\rho}_\lambda$ by $\tilde{\rho}_\lambda$ to make sense.*

We finally provide an application of Theorem 5.4.3. Under the additional constraint that the margin assumption is satisfied, we obtain a better rate.

Corollary 5.5.5 *Assume that the VB approximation is done on either $\mathcal{F}_1$, $\mathcal{F}_2$ or $\mathcal{F}_3$. Under Assumption A1 (Definition 5.5.1 page 133), and under Mammen and Tsybakov margin assumption, with $\lambda = \frac{2n}{C+2}$ and $\vartheta > 0$, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$,*

$$\left.\begin{array}{l} \int R\mathrm{d}\hat{\rho}_\lambda \\ \int R\mathrm{d}\tilde{\rho}_\lambda \end{array}\right\} \leq \bar{R} + \frac{(C+2)(C+1)}{2}\left\{\frac{d\log\frac{n}{\vartheta}}{n} + \frac{2d\vartheta}{n^2} + \frac{2}{\vartheta} - \frac{d}{\vartheta n} + \frac{2}{n}\log\frac{2}{\varepsilon}\right\} + \frac{\sqrt{d}2c(2C+1)}{n}.$$

The prior variance optimizing the bound is $\vartheta = d/(d + 2 + 2d/n)$, this choice or any constant instead will lead to a rate in $d\log(n)/n$. Note that the rate d/n is minimax-optimal in this context. This is, for example, a consequence of more general results in Lecué [2007] under a general form of the the margin assumption. See the Appendix for a proof.

5.5.4 Implementation and numerical results

For family $\mathcal{F}_2$ (mean field), the variational lower bound (5.5) equals

$$\mathcal{L}_{\lambda,\vartheta}(\mathbf{m},\boldsymbol{\sigma}) = -\frac{\lambda}{n}\sum_{i=1}^{n}\Phi\left(-Y_i\frac{X_i\mathbf{m}}{\sqrt{X_i\mathrm{diag}(\boldsymbol{\sigma}^2)X_i^t}}\right) - \frac{\boldsymbol{m}^T\boldsymbol{m}}{2\vartheta} + \frac{1}{2}\sum_{k=1}^{d}\left(\log\sigma_k^2 - \frac{\sigma_k^2}{\vartheta}\right),$$

while for family $\mathcal{F}_3$ (full covariance), it equals

$$\mathcal{L}_{\lambda,\vartheta}(\boldsymbol{m},\Sigma) = -\frac{\lambda}{n}\sum_{i=1}^{n}\Phi\left(-Y_i\frac{X_i\boldsymbol{m}}{\sqrt{X_i\Sigma X_i^t}}\right) - \frac{\boldsymbol{m}^T\boldsymbol{m}}{2\vartheta} + \frac{1}{2}\left(\log|\Sigma| - \frac{1}{\vartheta}\mathrm{tr}\Sigma\right).$$

Both functions are non-convex, but the multimodality of the latter may be more severe due to the larger dimension of $\mathcal{F}_3$. To address this issue, we recommend to use the reparametrisation of Opper and Archambeau [2009], which makes the dimension of the latter optimisation problem $\mathcal{O}(n)$; see Khan [2014] for a related approach. In both cases, we found that deterministic annealing to be a good approach to optimise such non-convex functions. We refer to Appendix B for more details on deterministic annealing and on our particular implementation.

We now compare the numerical performance of the mean field and full covariance VB approximations to the Gibbs posterior (as approximated by SMC, see Section 5.3.1) for the classification of standard datasets; see Table 5.1. We also include results for a kernel SVM (support vector machine); this comparison is not entirely fair, since SVM is a non-linear classifier, while all the other classifiers are linear. Still, except for the Glass dataset, the full covariance VB approximation performs as well or better than both SMC and SVM (while being much faster to compute, especially compared to SMC).

Interestingly, VB outperforms SMC in certain cases. This might be due to the fact that a VB approximation tends to be more concentrated around the mode than the Gibbs posterior it approximates. Mean field VB does not perform so well on certain datasets (e.g. Indian). This may be due either to the approximation family being too small, or to the corresponding optmisation problem being strongly multi-modal.

5.6 Application to classification under convexified loss

Compared to the previous section, the advantage of convex classification is that the corresponding variational approximation will amount to minimising a convex

Dataset	Covariates	Mean Field ($\mathcal{F}_2$)	Full cov. ($\mathcal{F}_3$)	SMC	SVM
Pima	7	31.0	21.3	22.3	30.4
Credit	60	32.0	33.6	32.0	32.0
DNA	180	23.6	23.6	23.6	20.4
SPECTF	22	08.0	06.9	08.5	10.1
Glass	10	34.6	19.6	23.3	4.7
Indian	11	48.0	25.5	26.2	26.8
Breast	10	35.1	1.1	1.1	1.7

Table 5.1: Comparison of misclassification rates (%).
Misclassification rates for different datasets and for the proposed approximations of the Gibbs posterior. The last column is the missclassification rate given by a kernel-SVM with radial kernel. The hyper-parameters are chosen by cross-validation.

function. This means that (a) the minimisation problem will be easier to deal with; and (b) we will be able to compute a bound for the integrated risk after a given number of steps of the minimisation procedure.

The setting is the same as in the previous section, except that for convenience we now take $\mathcal{Y} = \{-1, 1\}$, and the risk is based on the hinge loss,

$$r_n^H(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - Y_i < \theta, X_i >).$$

We will write R^H for the theoretical counterpart and $\bar{R}^H$ for its minimum in θ. We keep the superscript H in order to allow comparison with the risk R under the $0-1$ loss. We assume in this section that the X_i are uniformly bounded by a constant, $|X_i| < c_x$. Note that we do not require an assumption of the form (A1) to obtain the results of this section, as we rely directly on the Lipschitz continuity of the hinge risk.

5.6.1 Theoretical Results

Contrarily to the previous section, the risk is not bounded in θ, and we must specify a prior distribution for the Hoeffding assumption to hold.

Lemma 5.6.1 *Under an independent Gaussian prior π such that each component is $N(0, \vartheta^2)$, and for $\lambda < \frac{1}{c_x}\sqrt{\frac{n}{\vartheta^2}}$ and with bounded design $|X_{ij}| < c_x$, Hoeffding assumption* (5.1) *is satisfied with $f(\lambda, n) = \lambda^2/(4n) - \frac{1}{2}\log\left(1 - \frac{\vartheta^2\lambda^2 c_x^2}{2n}\right)$.*

The main impact of such a bound is that the prior variance cannot be taken too big relative to λ.

Corollary 5.6.2 *Assume that the VB approximation is done on either $\mathcal{F}_1$, $\mathcal{F}_2$ or $\mathcal{F}_3$. Take $\lambda = \frac{1}{c_x}\sqrt{\frac{n}{\vartheta^2}}$ and $\vartheta = \frac{1}{\sqrt{d}}$. For any $\varepsilon > 0$, with probability at least $1-\varepsilon$ we have simultaneously*

$$\left.\begin{array}{l} \int R^H \mathrm{d}\hat{\rho}_\lambda \\ \int R^H \mathrm{d}\tilde{\rho}_\lambda \end{array}\right\} \leq \overline{R}^H + \frac{c_x}{2}\sqrt{\frac{d}{n}}\log\frac{n}{d} + 2c_x\frac{d}{n} + \frac{1}{\sqrt{nd}}\left(\frac{c_x^2+1}{2c_x} + 2c_x\log\frac{2}{\epsilon}\right)$$

The oracle inequality in the above corollary enjoys the same rate of convergence as the equivalent result in the preceding section. In the following we link the two results.

Remark 5.6.1 *As stated in the beginning of the section we can use the estimator specified under the hinge loss to bound the excess risk of the 0-1 loss. We write $R^\star$ and $R^{H\star}$ the respective risk for their corresponding Bayes classifiers. From Zhang [2004] (section 3.3) we have the following inequality, linking the excess risk under the hinge loss and the $0-1$ loss,*

$$R(\theta) - R^\star \leq R^H(\theta) - R^{H\star}$$

for every $\theta \in \mathbb{R}^d$. By integrating with respect to $\tilde{\rho}^H$ (the VB approximation on any $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ of the Gibbs posterior for the hinge risk) and making use of Corollary 5.6.2 we have with high probability,

$$\tilde{\rho}^H\left(R(\theta)\right) - R^\star \leq \inf_{\theta\in\mathbb{R}^p} R^H(\theta) - R^{H\star} + \mathcal{O}\left(\sqrt{\frac{d}{n}}\log\left(\frac{n}{d}\right)\right).$$

5.6.2 Numerical application

We have motivated the introduction of the hinge loss as a convex upper bound. In the sequel we show that the resulting VB approximation also leads to a convex optimization problem. This has the advantage of opening a range of possible optimization algorithms [Nesterov, 2004]. In addition we are able to bound the error of the approximated measure after a fixed number of iterations (see Theorem 5.6.3).

Under the model $\mathcal{F}_1$ each individual risk is given by:

$$\rho_{m,\sigma}(r_i(\theta)) = (1-\Gamma_i m)\,\Phi\left(\frac{1-\Gamma_i m}{\sigma\|\Gamma_i\|_2}\right) + \sigma\|\Gamma_i\|\varphi\left(\frac{1-\Gamma_i m}{\sigma\|\Gamma_i\|_2}\right) := \Xi_i\left(\begin{pmatrix} m \\ \sigma \end{pmatrix}\right),$$

writting $\Gamma_i := Y_i X_i$.

Hence the lower bound to be maximized is given by

$$\mathcal{L}(m,\sigma) = -\frac{\lambda}{n}\left\{\sum_{i=1}^{n}(1-\Gamma_i m)\,\Phi\left(\frac{1-\Gamma_i m}{\sigma\|\Gamma_i\|_2}\right)+\sum_{i=1}^{n}\sigma\|\Gamma_i\|\varphi\left(\frac{1-\Gamma_i m}{\sigma\|\Gamma_i\|_2}\right)\right\} - \frac{\|m\|_2^2}{2\vartheta}+\frac{d}{2}\left(\log\sigma^2-\frac{\vartheta}{\sigma^2}\right).$$

It is easy to see that the function is convex in (m,σ), first note that the map

$$\Psi:\begin{pmatrix}x\\y\end{pmatrix}\mapsto x\Phi\left(\frac{x}{y}\right)+y\varphi\left(\frac{x}{y}\right),$$

is convex and note that we can write $\Xi_i\left(\begin{pmatrix}m\\\sigma\end{pmatrix}\right)=\Psi\left(A\begin{pmatrix}x\\y\end{pmatrix}+b\right)$ hence by composition of convex function with linear mappings we have the result. Similar reasoning could be held for the case $\mathcal{F}_2$ and $\mathcal{F}_3$, where in later the parametrization should be done in C such that $\Sigma = CC^t$. The bound is however not universally Lipschitz in σ, this impacts the optimization algorithms.

On the class of function $\mathcal{F}_0 = \left\{\Phi_{m,\frac{1}{n}}, m\in\mathbb{R}^d\right\}$, for which our Oracle inequalities still hold we could get faster numerical algorithms. The objective function has Lipschitz continuous derivatives and we would get a rate of $\frac{L}{(1+k)^2}$.

Other convex loss could be considered which could lead to convex optimization problems. For instance one could consider the exponential loss.

Dataset	**Covariates**	**Hinge loss**	**SMC**
Pima	7	21.8	22.3
Credit	60	27.2	32.0
DNA	180	4.2	23.6
SPECTF	22	19.2	08.5
Glass	10	26.12	23.3
Indian	11	26.2	25.5
Breast	10	0.5	1.1

Table 5.2: Comparison of misclassification rates (%).
Misclassification rates for different datasets and for the proposed approximations of the Gibbs posterior. The hyperparameters are chosen by cross-validation. This is to be compared to Table 5.1.

Theorem 5.6.3 *Assume that the VB approximation is done on $\mathcal{F}_1, \mathcal{F}_2$ or $\mathcal{F}_3$. Denote by $\tilde{\rho}_k(\mathrm{d}\theta)$ the VB approximated measure after the kth iteration of an optimal convex solver using the hinge loss. Take $\lambda = \frac{1}{c_x}\sqrt{\frac{n}{d}}$ and $\vartheta = \frac{1}{\sqrt{d}}$ then under the hypothesis of Corollary 5.6.2 with probability $1-\epsilon$*

$$\int R^H \mathrm{d}\tilde{\rho}_k \leq \overline{R}^H + \frac{LM}{\sqrt{1+k}} + \frac{c_x}{2}\sqrt{\frac{d}{n}}\log\frac{n}{d} + 2c_x\frac{d}{n} + \frac{1}{\sqrt{nd}}\left(\frac{c_x^2+1}{2c_x} + 2c_x\log\frac{2}{\epsilon}\right)$$

where L is the Lipschitz coefficient on a ball of radius M of the objective function maximized in VB.

From Theorem 5.6.3 we can compute the number of iterations to get a given level of error at a given probability.

We find that on average the misclassification error (Table 5.2) is lower than for the 0-1 loss where we have no guaranties that the maximum is attained.

5.7 Application to ranking

5.7.1 Preliminaries

In this section we take $\mathcal{Y} = \{0,1\}$ and consider again linear classifiers: $\Theta = \mathcal{X} = \mathbb{R}^d$, $f_\theta(x) = \mathbf{1}_{\langle\theta,x\rangle\geq 0}$. We consider however a different criterion: in ranking, not only we want to classify well an object x, but we want to make sure that given two different objects, the one that is more likely to correspond to a label 1 will be assigned a larger score through the function f_θ. A usual way to measure this is to introduce the risk function

$$R(\theta) = \mathbb{P}[(Y_1 - Y_2)(f_\theta(X_1) - f_\theta(X_2)) < 0]$$

and the empirical risk

$$r_n(\theta) = \frac{1}{n(n-1)}\sum_{1\leq i\neq j\leq n}\mathbf{1}_{\{(Y_i-Y_j)(f_\theta(X_i)-f_\theta(X_j))<0\}}.$$

Then, again, we recall classical results.

Lemma 5.7.1 *The Hoeffding-type assumption is satisfied with $f(\lambda, n) = \frac{\lambda^2}{n-1}$.*

The variant of the margin assumption adapted to ranking was established by Robbiano [2013] and Ridgway [2015].

Lemma 5.7.2 *Assume the following margin assumption:*

$$\mathbb{E}[(\mathbf{1}_{[f_\theta(X_1)-f_\theta(X_2)][Y_1-Y_2]<0} - \mathbf{1}_{[f_{\overline{\theta}}(X_1)-f_{\overline{\theta}}(X_2)][Y_1-Y_2]<0})^2] \leq C[R(\theta) - \overline{R}].$$

Then Bernstein assumption (5.2) *is satisfied with $g(\lambda, n) = \frac{C\lambda^2}{n-1-4\lambda}$.*

We still consider a Gaussian prior

$$\pi(\mathrm{d}\theta) = \prod_{i=1}^{d} \varphi(\theta_i; 0, \vartheta^2)\mathrm{d}\theta_i$$

and the approximation families will be the same as in Section 5.5: $\mathcal{F}_1 = \{\Phi_{\mathbf{m},\sigma^2}, \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in \mathbb{R}_+^*\}$, $\mathcal{F}_2 = \{\Phi_{\mathbf{m},\boldsymbol{\sigma}^2}, \mathbf{m} \in \mathbb{R}^d, \boldsymbol{\sigma}^2 \in (\mathbb{R}_+^*)^d\}$ and $\mathcal{F}_3 = \{\Phi_{\mathbf{m},\Sigma}, \mathbf{m} \in \mathbb{R}^d, \Sigma \in \mathcal{S}^{d+}\}$.

5.7.2 Theoretical study

Here again, we start with the empirical bound.

Corollary 5.7.3 *For any $\varepsilon > 0$, with probability at least $1-\varepsilon$ we have, for any $\boldsymbol{m} \in \mathbb{R}^d$, $\sigma^2 \in (\mathbb{R}_+)^d$,*

$$\int R\mathrm{d}\Phi_{\mathbf{m},\boldsymbol{\sigma}^2} \leq \int r_n \mathrm{d}\Phi_{\mathbf{m},\boldsymbol{\sigma}^2} + \frac{\lambda}{n-1} + \frac{\sum_{j=1}^{d}\left[\frac{1}{2}\log\left(\frac{\vartheta^2}{\sigma_i^2}\right) + \frac{\sigma_i^2}{\vartheta^2}\right] + \frac{\|\boldsymbol{m}\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{1}{\varepsilon}\right)}{\lambda}.$$

In order to derive a theoretical bound, we introduce the following variant of Assumption A1.

Definition 5.7.1 *We say that Assumption A2 is satisfied when there is a constant $c > 0$ such that, for any $(\theta, \theta') \in \Theta^2$ with $\|\theta\| = \|\theta'\| = 1$, $\mathbb{P}(\langle X_1 - X_2, \theta\rangle \langle X_1 - X_2, \theta'\rangle < 0) \leq c\|\theta - \theta'\|$.*

Assumption A2 is satisfied as soon as $(X_1 - X_2)/\|X_1 - X_2\|$ has a bounded density on the unit sphere.

Corollary 5.7.4 *Use either $\mathcal{F}_1$, $\mathcal{F}_2$ or $\mathcal{F}_3$. Take $\lambda = \sqrt{\frac{d(n-1)}{2}}$ and $\vartheta = 1$. Under (A2), for any $\varepsilon > 0$, with probability at least $1-\varepsilon$,*

$$\left.\begin{array}{l}\int R\mathrm{d}\hat{\rho}_\lambda \\ \int R\mathrm{d}\tilde{\rho}_\lambda\end{array}\right\} \leq \overline{R} + \sqrt{\frac{2d}{n-1}}\left(1 + \frac{1}{2}\log\left(2d(n-1)\right)\right) + \frac{c\sqrt{2}}{\sqrt{n-1}} + \frac{2\sqrt{2}\log\left(\frac{2\mathrm{e}}{\varepsilon}\right)}{\sqrt{(n-1)d}}.$$

Finally, under an additional margin assumption, we have:

Corollary 5.7.5 *Under Assumption A2 and the margin assumption of Lemma (5.7.2), for $\lambda = \frac{n-1}{C+5}$ and $\vartheta > 0$, for any $\varepsilon > 0$, with probability at least $1-\varepsilon$,*

$$\left.\begin{array}{l}\int R\mathrm{d}\hat{\rho}_\lambda \\ \int R\mathrm{d}\tilde{\rho}_\lambda\end{array}\right\} \leq \bar{R} + \frac{(C+5)(C+1)}{2}\left\{\frac{d\log\frac{n}{\vartheta}}{n-1} + \frac{2d\vartheta}{n(n-1)} + \frac{2}{\vartheta} - \frac{d}{\vartheta n - 1} + \frac{2}{n-1}\log\frac{2}{\varepsilon}\right\} + \frac{\sqrt{d}4c(C+1)}{n}.$$

The prior variance optimizing the bound is $\vartheta = d/(d + 2 + 2d/n)$. The proof is similar to the ones of Corollaries 5.5.4, 5.5.5 and 5.7.4.

As in the case of classification, ranking under an AUC loss can be done by replacing the indicator function by the corresponding upper bound given by an hinge loss. In this case we can derive similar results as for the convexified classification in particular we can get a convex minimization problem and obtain result without requiring assumption (A2).

5.7.3 Algorithms and numerical results

As an illustration we focus here on family $\mathcal{F}_2$ (mean field). In this case the VB objective to maximize is given by:

$$\mathcal{L}(\boldsymbol{m}, \sigma^2) = -\frac{\lambda}{n_+ n_-} \sum_{i:y_i=1, j:y_j=0} \Phi\left(-\frac{\Gamma_{ij} m}{\sqrt{\sum_{k=1}^d (\gamma_{ij}^k)^2 \sigma_k^2}}\right) - \frac{\|\boldsymbol{m}\|_2^2}{2\vartheta} + \frac{1}{2}\sum_{k=1}^d \left[\log \sigma_k^2 - \frac{\sigma_k^2}{\vartheta}\right], \tag{5.7}$$

where $\Gamma_{ij} = X_i - X_j$, and where $(\gamma_{ij}^k)_k$ are the elements of Γ.

This function is expensive to compute, as it involves $n_+ n_-$ terms, the computation of which is $\mathcal{O}(p)$.

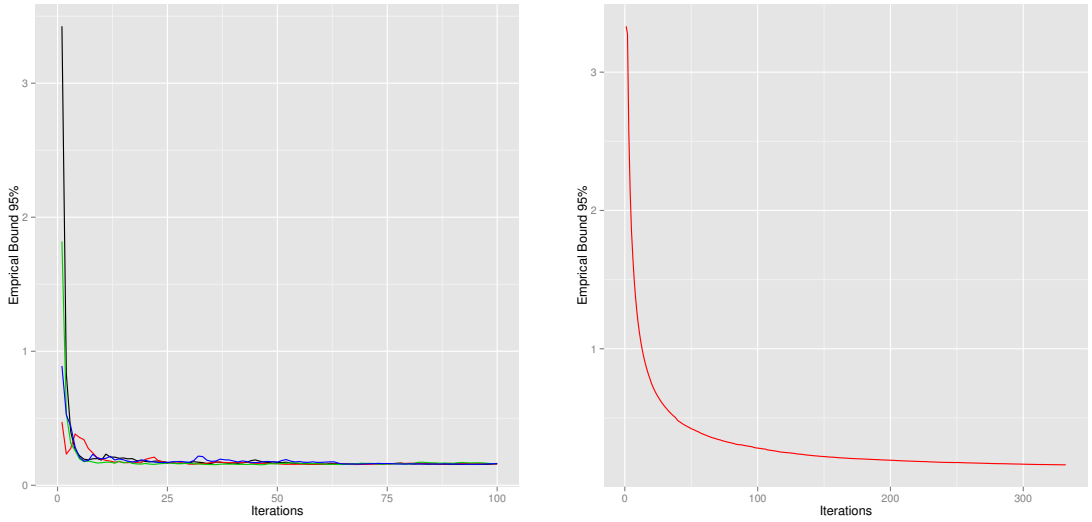
We propose to use a stochastic gradient descent in the spirit of Hoffman et al. [2013]. The model we consider is not in an exponential family, meaning we cannot use the trick developed by these authors. We propose instead to use a standard descent.

The idea is to replace the gradient by an unbiased version based on a batch of size B as described in Algorithm 22 in the Appendix. Robbins and Monro [1951] show that for a step-size $(\lambda_t)_t$ such that $\sum_t \lambda_t^2 < \infty$ and $\sum_t \lambda_t = \infty$ the algorithm converges to a local optimum.

In our case we propose to sample pairs of data with replacement and use the unbiased version of the derivative of the risk component. We use a simple gradient descent without any curvature information. One could also use recent research on stochastic quasi Newton-Raphson [Byrd et al., 2014].

For illustration, we consider a small dataset (Pima), and a larger one (Adult). The latter is already quite challenging with $n_+ n_- = 193,829,520$ pairs to compare. In both cases with different size of batches convergence is obtained with a few iterations only and leads to acceptable bounds.

In Figure 5.1 we show the empirical bound on the AUC risk as a function of the iteration of the algorithm, for several batch sizes. The bound is taken for 95% probability, the batch sizes are taken to be $B = 1, 10, 20, 50$ for the Pima dataset, and 50 for the Adult dataset. The figure shows an additional feature of VB approximation in the context of Gibbs posterior: namely the possibility of



(a) Pima (b) adult

Figure 5.1: Error bound at each iteration, stochastic descent, Pima and Adult datasets.

Stochastic VB with fixed temperature $\lambda = 100$ for Pima and $\lambda = 1000$ for adult. The left panel shows several curves that correspond to different batch sizes; these curves are hard to distinguish. The right panel is for a batch size of 50. The adult dataset has $n = 32556$ observation and $n_+ n_- = 193829520$ possible pairs. The convergence is obtained in order of seconds. The bounds are the empirical bounds obtained in Corollary 5.7.3 for a probability of 95%.

computing the empirical upper bound given by Corollary 5.7.3. That is we can check the quality of the bound at each iteration of the algorithm, or for different values of the hyperparameters.

5.8 Application to matrix completion

The matrix completion problem has received increasing attention recently, partly due to spectacular theoretical results [Candès and Tao, 2010], and to challenging applications like the Netflix challenge [Bennett and Lanning, 2007]. In the perspective of this chapter, the specific interest of this application is twofold. First, this is a case where the family of approximations is not parametric, but rather of the form (5.3), i.e. the family of products of independent components. Then, there is no known theoretical result for the Gibbs estimator in the considered model, yet we can still directly bound the loss induced by the variational approximation.

We observe i.i.d. pairs $((X_i, Y_i))_{i=1}^n$ where $X_i \in \{1, \dots, m_1\} \times \{1, \dots, m_2\}$, and we assume that there is a $m_1 \times m_2$-matrix M such that $Y_i = M_{X_i} + \varepsilon_i$ and the ε_i are centred. Assuming that X_i is uniform on $\{1, \dots, m_1\} \times \{1, \dots, m_2\}$, that $f_\theta(X_i) = \theta_{X_i}$, and taking the quadratic risk, $R(\theta) = \mathbb{E}\left[(Y_i - \theta_{X_i})^2\right]$, we have that

$$R(\theta) - \overline{R} = \frac{1}{m_1 m_2} \|\theta - M\|_F^2$$

where $\|\cdot\|_F$ stands for the Frobenius norm.

A common way to parametrise the problem is

$$\Theta = \{\theta = UV^T, U \in \mathbb{R}^{m_1 \times K}, V \in \mathbb{R}^{m_2 \times K}\}$$

where K is large; e.g. $K = \min(m_1, m_2)$. Following Salakhutdinov and Mnih [2008], we define the following prior distribution: $U_{\cdot,j} \sim \mathcal{N}(0, \gamma_j I)$, $V_{\cdot,j} \sim \mathcal{N}(0, \gamma_j I)$ where the γ_j's are i.i.d. from an inverse gamma distribution, $\gamma_j \sim \mathcal{IG}(a, b)$.

Note that VB algorithms were used in this context by Lim and Teh [2007] (with a slightly simpler prior however: the γ_j's are fixed rather than random). Since then, this prior and variants were used in several papers [e.g. Lawrence and Urtasun, 2009; Zhou et al., 2010]. Until now, no theoretical results were proved up to our knowledge. Two papers prove minimax-optimal rates for slightly modified estimators (by truncation), for which efficient algorithms are unknown [Mai and Alquier, 2015; Suzuki, 2014]. However, using Theorems 5.4.2 and 5.4.3 we are able to prove the following: *if* there is a PAC-Bayesian bound leading to a rate for $\hat{\rho}_\lambda$ in this context, then the same rate holds for $\tilde{\rho}_\lambda$. In other words: if someone proves the conjecture that the Gibbs estimator is minimax-optimal (up to log terms) in this context, then the VB approximation will enjoy automatically the same property.

We propose the following approximation:

$$\mathcal{F} = \left\{ \rho(\mathrm{d}(U, V)) = \prod_{i=1}^{m_1} u_i(\mathrm{d}U_{i,\cdot}) \prod_{j=1}^{m_2} v_j(\mathrm{d}V_{j,\cdot}) \right\}.$$

Theorem 5.8.1 *Assume that $M = UV^T$ with $|U_{i,k}|, |V_{j,k}| \leq C$. Assume that $\mathrm{rank}(M) = r$ so that we can assume that $U_{\cdot,r+1} = \cdots = U_{\cdot,K} = V_{\cdot,r+1} = \cdots = V_{\cdot,K} = 0$ (note that the prior π does not depend on the knowledge of r though). Choose the prior distribution on the hyper-parameters γ_j as inverse gamma $\mathrm{Inv}{-}\Gamma(a, b)$ with $b \leq 1/[2\beta(m_1 \vee m_2)\log(2K(m_1 \vee m_2))]$. Then there is a constant $\mathcal{C}(a, C)$ such that, for any $\beta > 0$,*

$$\inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_\beta) \leq \mathcal{C}(a, C) \left\{ r(m_1 + m_2) \log\left[\beta b (m_1 + m_2) K\right] + \frac{1}{\beta} \right\}.$$

See the Appendix for a proof.

For instance, in Theorem 5.4.3, in classification and ranking we had λ, $\lambda - g(\lambda, n)$ and $\lambda + g(\lambda, n)$ of order $\mathcal{O}(n)$. In this case we would have:

$$\frac{2}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{F}} \mathcal{K}\left(\rho, \pi_{\frac{\lambda + g(\lambda,n)}{2}}\right) = \mathcal{O}\left(\frac{\mathcal{C}(a, C) r (m_1 + m_2) \log\left[nb(m_1 + m_2)K\right]}{n}\right),$$

and note that in this context it is known that the minimax rate is at least $r(m_1 + m_2)/n$ [Koltchinskii et al., 2011].

5.8.1 Algorithm

As already mentioned, the approximation family is not parametric in this case, but rather of type mean field. The corresponding VB algorithm amounts to iterating equation (5.4), which takes the following form in this particular case:

$$u_j(\mathrm{d}U_{j,\cdot}) \propto \exp\left\{-\frac{\lambda}{n}\sum_i \mathbb{E}_{V,U_{-j}}\left[(Y_{X_i} - (UV^T)_{X_i})^2\right] - \sum_{k=1}^{K} \mathbb{E}_{\gamma_j}\left[\frac{1}{2\gamma_k}\right] U_{jk}^2\right\}$$

$$v_j(\mathrm{d}V_{j,\cdot}) \propto \exp\left\{-\frac{\lambda}{n}\sum_i \mathbb{E}_{V_{-j},U}\left[(Y_{X_i} - (UV^T)_{X_i})^2\right] - \sum_{k=1}^{K} \mathbb{E}_{\gamma_j}\left[\frac{1}{2\gamma_k}\right] V_{jk}^2\right\}$$

$$p(\gamma_k) \propto \exp\left\{-\frac{1}{2\gamma_k}\left(\sum_j \mathbb{E}_U U_{kj}^2 + \sum_i \mathbb{E}_V V_{ik}^2\right) + (\alpha + 1)\log\frac{1}{\gamma_k} - \frac{\beta}{\gamma_k}\right\}$$

where the expectations are taken with respect to the thus defined variational approximations. One recognises Gaussian distributions for the first two, and an inverse Gamma distribution for the third. We refer to Lim and Teh [2007] for more details on this algorithm and for a numerical illustration.

5.9 Discussion

We showed in several important scenarios that approximating a Gibbs posterior through VB (Variational Bayes) techniques does not deteriorate the rate of convergence of the corresponding procedure. We also described practical algorithms for fast computation of these VB approximations, and provided empirical bounds that may be computed from the data to evaluate the performance of the so-obtained VB-approximated procedure. We believe these results provide a strong incentive to recommend VB as the default approach to approximate Gibbs posteriors, in lieu of Monte Carlo methods.

We hope to extend our results to other applications beyond those discussed in this chapter, such as regression. One technical difficulty with regression is that

the risk function is not bounded, which makes our approach a bit less direct to apply. In many papers on PAC-Bayesian bounds for regression, the noise can be unbounded (usually, it is assumed to be sub-exponential), but one assumes that the predictors are bounded, see e.g. Alquier and Biau [2013]. However, using the robust loss function of Audibert and Catoni, it is possible to relax this assumption [Audibert and Catoni, 2011; Catoni, 2012]. This requires a more technical analysis, which we leave for further work.

Appendix

5.A Proofs

5.A.1 Preliminary remarks

We start by a general remark. Let h be a function $\Theta \to \mathbb{R}_+$ with $\int \exp[-h(\theta)]\pi(\mathrm{d}\theta) < \infty$. Let us put

$$\pi[h](\mathrm{d}\theta) = \frac{\exp[-h(\theta)]}{\int \exp[-h(\theta')]\pi(\mathrm{d}\theta')}\pi(\mathrm{d}\theta).$$

Direct calculation yields, for any $\rho \ll \pi$ with $\int h\mathrm{d}\rho < \infty$,

$$\mathcal{K}(\rho, \pi[h]) = \lambda \int h\mathrm{d}\rho + \mathcal{K}(\rho, \pi) + \log \int \exp(-h)\mathrm{d}\pi.$$

Two well known consequences are

$$\pi[h] = \arg \min_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \int h\mathrm{d}\rho + \mathcal{K}(\rho, \pi) \right\},$$
$$-\log \int \exp(-h)\mathrm{d}\pi = \min_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \int h\mathrm{d}\rho + \mathcal{K}(\rho, \pi) \right\}.$$

We will use these inequalities many times in the followings. The most frequent application will be with $h(\theta) = \lambda r_n(\theta)$ (in this case $\pi[\lambda r_n] = \hat{\rho}_\lambda$) or $h(\theta) = \pm\lambda[r_n(\theta) - R(\theta)]$, the first case leads to

$$\mathcal{K}(\rho, \hat{\rho}_\lambda) = \lambda \int r_n\mathrm{d}\rho + \mathcal{K}(\rho, \pi) + \log \int \exp(-\lambda r_n)\mathrm{d}\pi, \qquad (5.8)$$

$$\hat{\rho}_\lambda = \arg \min_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int r_n \mathrm{d}\rho + \mathcal{K}(\rho, \pi) \right\}, \tag{5.9}$$

$$-\log \int \exp(-\lambda r_n) \mathrm{d}\pi = \min_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int r_n \mathrm{d}\rho + \mathcal{K}(\rho, \pi) \right\}. \tag{5.10}$$

We will use (5.8), (5.9) and (5.10) several times in this appendix.

5.A.2 Proof of the theorems in Subsection 5.4.1

Proof of Theorem 5.4.1. This proof follows the standard PAC-Bayesian approach (see Catoni [2007]). Apply Fubini's theorem to the first inequality of (5.1):

$$\mathbb{E} \int \exp\left\{ \lambda [R(\theta) - r_n(\theta)] - f(\lambda, n) \right\} \pi(\mathrm{d}\theta) \leq 1$$

then apply the preliminary remark with $h(\theta) = \lambda[r_n(\theta) - R(\theta)]$:

$$\mathbb{E} \exp \left\{ \sup_{\rho \in \mathcal{M}_+^1(\Theta)} \int \lambda [R(\theta) - r_n(\theta)] \rho(\mathrm{d}\theta) - \mathcal{K}(\rho, \pi) - f(\lambda, n) \right\} \leq 1.$$

Multiply both sides by ε and use $\mathbb{E}[\exp(U)] \geq \mathbb{P}(U > 0)$ for any U to obtain:

$$\mathbb{P} \left[\sup_{\rho \in \mathcal{M}_+^1(\Theta)} \int \lambda [R(\theta) - r_n(\theta)] \rho(\mathrm{d}\theta) - \mathcal{K}(\rho, \pi) - f(\lambda, n) + \log(\varepsilon) > 0 \right] \leq \varepsilon.$$

Then consider the complementary event:

$$\mathbb{P} \left[\forall \rho \in \mathcal{M}_+^1(\Theta), \quad \lambda \int R \mathrm{d}\rho \leq \lambda \int r_n \mathrm{d}\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{1}{\varepsilon}\right) \right] \geq 1 - \varepsilon.$$

□

Proof of Theorem 5.4.2. Using the same calculations as above, we have, with probability at least $1 - \varepsilon$, simultaneously for all $\rho \in \mathcal{M}_+^1(\Theta)$,

$$\lambda \int R \mathrm{d}\rho \leq \lambda \int r_n \mathrm{d}\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right) \tag{5.11}$$

$$\lambda \int r_n \mathrm{d}\rho \leq \lambda \int R \mathrm{d}\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right). \tag{5.12}$$

We use (5.11) with $\rho = \hat{\rho}_\lambda$ and (5.9) to get

$$\lambda \int R \mathrm{d}\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int r_n \mathrm{d}\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right) \right\}$$

and plugging (5.12) into the right-hand side, we obtain

$$\lambda \int R \mathrm{d}\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int R \mathrm{d}\rho + 2f(\lambda, n) + 2\mathcal{K}(\rho, \pi) + 2\log\left(\frac{2}{\varepsilon}\right) \right\}.$$

Now, we work with $\tilde{\rho}_\lambda = \arg\min_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \hat{\rho}_\lambda)$. Plugging (5.8) into (5.11) we get, for any ρ,

$$\lambda \int R \mathrm{d}\rho \leq f(\lambda, n) + \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp(-\lambda r_n) \mathrm{d}\pi + \log\left(\frac{2}{\varepsilon}\right).$$

By definition of $\tilde{\rho}_\lambda$, we have:

$$\lambda \int R \mathrm{d}\tilde{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ f(\lambda, n) + \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp(-\lambda r_n) \mathrm{d}\pi + \log\left(\frac{2}{\varepsilon}\right) \right\}$$

and, using (5.8) again, we obtain:

$$\lambda \int R \mathrm{d}\tilde{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int r_n \mathrm{d}\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right) \right\}.$$

We plug (5.12) into the right-hand side to obtain:

$$\lambda \int R \mathrm{d}\tilde{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int R \mathrm{d}\rho + 2f(\lambda, n) + 2\mathcal{K}(\rho, \pi) + 2\log\left(\frac{2}{\varepsilon}\right) \right\}.$$

This proves the second inequality of the theorem. In order to prove the claim

$$\mathcal{B}_\lambda(\mathcal{F}) = \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}),$$

note that

$$\begin{aligned}
\mathcal{B}_\lambda(\mathcal{F}) &= \inf_{\rho \in \mathcal{F}} \left\{ \int R \mathrm{d}\rho + \frac{2f(\lambda, n)}{\lambda} + \frac{2\mathcal{K}(\rho, \pi)}{\lambda} + \frac{2\log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\} \\
&= \inf_{\rho \in \mathcal{F}} \left\{ -\frac{2}{\lambda} \log \int \exp\left(-\frac{\lambda}{2} R\right) \mathrm{d}\pi + \frac{2f(\lambda, n)}{\lambda} + \frac{2\mathcal{K}(\rho, \pi_{\frac{\lambda}{2}})}{\lambda} + \frac{2\log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\} \\
&= -\frac{2}{\lambda} \log \int \exp\left(-\frac{\lambda}{2} R\right) \mathrm{d}\pi + \frac{2f(\lambda, n)}{\lambda} + \frac{2\log\left(\frac{2}{\varepsilon}\right)}{\lambda} + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}) \\
&= \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}).
\end{aligned}$$

This ends the proof. □

5.A.3 Proof of Theorem 5.4.3 (Subsection 5.4.2)

Proof of Theorem 5.4.3. As in the proof of Theorem 5.4.1, we apply Fubini, then (5.10) to the first inequality of (5.2) to obtain

$$\mathbb{E}\exp\left\{\sup_{\rho}\int\left[\lambda[R(\theta)-\overline{R}]-\lambda[r_n(\theta)-\overline{r}_n]-g(\lambda,n)[R(\theta)-\overline{R}]\right]\rho(\mathrm{d}\theta)-\mathcal{K}(\rho,\pi)\right\}\leq 1$$

and we multiply both sides by $\varepsilon/2$ to get

$$\mathbb{P}\left\{\sup_{\rho}\left[[\lambda-g(\lambda,n)]\left[\int R\mathrm{d}\rho-\overline{R}\right]\geq\lambda\left[\int r_n\mathrm{d}\rho-\overline{r}_n\right]+\mathcal{K}(\rho,\pi)+\log\left(\frac{2}{\varepsilon}\right)\right]\right\}\leq\frac{\varepsilon}{2}. \tag{5.13}$$

We now consider the second inequality in (5.2):

$$\mathbb{E}\exp\left\{\lambda[r_n(\theta)-\overline{r}_n]-\lambda[R(\theta)-\overline{R}]-g(\lambda,n)[R(\theta)-\overline{R}]\right\}\leq 1.$$

The same derivation leads to

$$\mathbb{P}\left\{\sup_{\rho}\left[[\lambda-g(\lambda,n)]\left[\int r_n\mathrm{d}\rho-\overline{r}_n\right]\geq\lambda\left[\int R\mathrm{d}\rho-\overline{R}\right]+\mathcal{K}(\rho,\pi)+\log\left(\frac{2}{\varepsilon}\right)\right]\right\}\leq\frac{\varepsilon}{2}. \tag{5.14}$$

We combine (5.13) and (5.14) by a union bound argument, and we consider the complementary event: with probability at least $1-\varepsilon$, simultaneously for all $\rho\in\mathcal{M}_+^1(\Theta)$,

$$[\lambda-g(\lambda,n)]\left[\int R\mathrm{d}\rho-\overline{R}\right]\leq\lambda\left[\int r_n\mathrm{d}\rho-\overline{r}_n\right]+\mathcal{K}(\rho,\pi)+\log\left(\frac{2}{\varepsilon}\right), \tag{5.15}$$

$$\lambda\left[\int r_n\mathrm{d}\rho-\overline{r}_n\right]\leq[\lambda+g(\lambda,n)]\left[\int R\mathrm{d}\rho-\overline{R}\right]+\mathcal{K}(\rho,\pi)+\log\left(\frac{2}{\varepsilon}\right). \tag{5.16}$$

We now derive consequences of these two inequalities (in other words, we focus on the event where these two inequalities are satisfied). Using (5.9) in (5.15) yields

$$[\lambda-g(\lambda,n)]\left[\int R\mathrm{d}\hat{\rho}_\lambda-\overline{R}\right]\leq\inf_{\rho\in\mathcal{M}_+^1(\Theta)}\left\{\lambda\left[\int r_n\mathrm{d}\rho-\overline{r}_n\right]+\mathcal{K}(\rho,\pi)+\log\left(\frac{2}{\varepsilon}\right)\right\}.$$

We plug (5.16) into the right-hand side to obtain:

$$\begin{aligned}&[\lambda-g(\lambda,n)]\left[\int R\mathrm{d}\hat{\rho}_\lambda-\overline{R}\right]\\&\qquad\leq\inf_{\rho\in\mathcal{M}_+^1(\Theta)}\left\{[\lambda+g(\lambda,n)]\left[\int R\mathrm{d}\rho-\overline{R}\right]+2\mathcal{K}(\rho,\pi)+2\log\left(\frac{2}{\varepsilon}\right)\right\}.\end{aligned}$$

Now, we work with $\tilde{\rho}_\lambda$. Plugging (5.8) into (5.13) we get

$$[\lambda - g(\lambda, n)] \left[\int R \mathrm{d}\rho - \overline{R} \right] \leq \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp[-\lambda(r_n - \overline{r}_n)] \mathrm{d}\pi + \log \left(\frac{2}{\varepsilon} \right).$$

By definition of $\tilde{\rho}_\lambda$, we have:

$$[\lambda - g(\lambda, n)] \left[\int R \mathrm{d}\tilde{\rho}_\lambda - \overline{R} \right] \leq \inf_{\rho \in \mathcal{F}} \left\{ \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp[-\lambda(r_n - \overline{r}_n)] \mathrm{d}\pi + \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

Then, apply (5.8) again to get:

$$[\lambda - g(\lambda, n)] \left[\int R \mathrm{d}\tilde{\rho}_\lambda - \overline{R} \right] \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int (r_n - \overline{r}_n) \mathrm{d}\rho + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

Plug (5.16) into the right-hand side to get

$$[\lambda - g(\lambda, n)] \left[\int R \mathrm{d}\tilde{\rho}_\lambda - \overline{R} \right] \leq \inf_{\rho \in \mathcal{F}} \left\{ [\lambda + g(\lambda, n)] \int (R - \overline{R}) \mathrm{d}\rho + 2\mathcal{K}(\rho, \pi) + 2 \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

□

5.A.4 Proofs of Section 5.5

Proof of Lemma 5.5.1. Combine Theorem 2.1 p. 25 and Lemma 2.2 p. 27 in Boucheron et al. [2013]. □

Proof of Lemma 5.5.2. Apply Theorem 2.10 in Boucheron et al. [2013], and plug the margin assumption. □

Proof of Corollary 5.5.4. We remind that thanks to (5.6) it is enough to prove the claim for $\mathcal{F}_1$. We apply Theorem 5.4.2 to get:

$$\begin{aligned}
\mathcal{B}_\lambda(\mathcal{F}_1) &= \inf_{(\mathbf{m}, \sigma^2)} \left\{ \int R \mathrm{d}\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{n} + 2 \frac{\mathcal{K}(\Phi_{\mathbf{m}, \sigma^2}, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\} \\
&= \inf_{(m, \sigma^2)} \left\{ \int R \mathrm{d}\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{n} + 2 \frac{d \left[\frac{1}{2} \log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2} \right] + \frac{\|\boldsymbol{m}\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}.
\end{aligned}$$

Note that the minimizer of R, $\overline{\theta}$, is not unique (because $f_\theta(x)$ does not depend on $\|\theta\|$) and we can chose it in such a way that $\|\overline{\theta}\| = 1$. Then

$$\begin{aligned} R(\theta) - \overline{R} &= \mathbb{E}\left[\mathbf{1}_{\langle\theta,X\rangle Y<0} - \mathbf{1}_{\langle\overline{\theta},X\rangle Y<0}\right] \leq \mathbb{E}\left[\mathbf{1}_{\langle\theta,X\rangle\langle\overline{\theta},X\rangle<0}\right] \\ &= \mathbb{P}\left(\langle\theta,X\rangle\left\langle\overline{\theta},X\right\rangle<0\right) \leq c\left\|\frac{\theta}{\|\theta\|} - \overline{\theta}\right\| \leq 2c\|\theta-\overline{\theta}\|. \end{aligned}$$

So:

$$\begin{aligned} \mathcal{B}_\lambda(\mathcal{F}_1) \leq \overline{R} + \inf_{(\mathbf{m},\sigma^2)}\Bigg\{ & 2c\int\|\theta-\overline{\theta}\|\Phi_{\mathbf{m},\sigma^2}(\mathrm{d}\theta) \\ & + \frac{\lambda}{n} + 2\frac{d\left[\frac{1}{2}\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2}\right] + \frac{\|\boldsymbol{m}\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda}\Bigg\}. \end{aligned}$$

We now restrict the infimum to distributions ν such that $\boldsymbol{m} = \overline{\theta}$:

$$\mathcal{B}(\mathcal{F}_1) \leq \overline{R} + \inf_{\sigma^2}\left\{2c\sqrt{d}\sigma + \frac{\lambda}{n} + \frac{d\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{2d\sigma^2}{\vartheta^2} + \frac{2}{\vartheta^2} - d + 2\log\left(\frac{2}{\varepsilon}\right)}{\lambda}\right\}.$$

We put $\sigma = \frac{1}{2\lambda}$ and substitute $\frac{1}{\sqrt{d}}$ for ϑ to get

$$\mathcal{B}(\mathcal{F}_1) \leq \overline{R} + \frac{\lambda}{n} + \frac{c\sqrt{d} + d\log(4\frac{\lambda^2}{d}) + \frac{d^2}{2\lambda^2} + d + 2\log\left(\frac{2}{\varepsilon}\right)}{\lambda}.$$

Substitute $\sqrt{nd}$ for λ to get the desired result. □

Proof of Corollary 5.5.5. We apply Theorem 5.4.3:

$$\begin{aligned} & \int(R-\overline{R})\mathrm{d}\tilde{\rho}_\lambda \\ & \leq \inf_{\mathbf{m},\sigma^2}\left\{\frac{\lambda+g(\lambda,n)}{\lambda-g(\lambda,n)}\int(R-\bar{R})\mathrm{d}\Phi_{\mathbf{m},\sigma^2} + \frac{1}{\lambda-g(\lambda,n)}\left(2\mathcal{K}(\Phi_{\mathbf{m},\sigma^2},\pi) + 2\log\frac{2}{\epsilon}\right)\right\} \end{aligned}$$

where $\lambda < \frac{2n}{C+1}$. Computations similar to those in the the proof of Corollary 5.5.4 lead to

$$\begin{aligned} \int R d\tilde{\rho}_\lambda \leq \overline{R} + \inf_{\mathbf{m},\sigma^2}\Bigg\{ & 2c\frac{\lambda+g(\lambda,n)}{\lambda-g(\lambda,n)}\int\|\theta-\overline{\theta}\|\Phi_{\mathbf{m},\sigma^2}(\mathrm{d}\theta) \\ & + 2\frac{\sum_{j=1}^d\left[\frac{1}{2}\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2}\right] + \frac{\|\boldsymbol{m}\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda-g(\lambda,n)}\Bigg\}. \end{aligned}$$

taking $\boldsymbol{m} = \bar{\theta}$ and $\lambda = \frac{2n}{C+2}$, we get the result. □

5.A.5 Proofs of Section 5.6

Proof of Lemma 5.6.1. For fixed θ we can upper bound the individual risk such that:

$$0 \leq \max(0, 1 - < \theta, X_i > Y_i) \leq 1 + | < \theta, X_i > |$$

such that we can apply Hoeffding's inequality conditionally on X_i and fixed θ.

We get,

$$\begin{aligned}\mathbb{E}\left[\exp\left(\lambda(R^H - r_n^H)\right)|X_1, \cdots, X_n\right] &\leq \exp\left\{\frac{\lambda^2}{8n^2}\sum_{i=1}^{n}(1 + | < \theta, X_i > |)^2\right\} \\ &\leq \exp\left\{\frac{\lambda^2}{4n} + \frac{\lambda^2 c_x^2}{4n}\|\theta\|^2\right\}\end{aligned}$$

where the last inequality stems from the fact that $(a+b)^2 \leq 2(a^2+b^2)$ and the fact that we have supposed the X_i to be bounded. We can take the expectation of this term with respect to the X_i's and with respect to our Gaussian prior.

$$\begin{aligned}\pi\left\{\mathbb{E}\left[\exp\left(\lambda(R^H - r_n^H)\right)\right]\right\} &\leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{(2\pi)^{\frac{d}{2}}\sqrt{\vartheta^2}}\int \exp\left(\frac{\lambda^2 c_x^2}{4n}\|\theta\|^2 - \frac{1}{2\vartheta^2}\|\theta\|^2\right)\mathrm{d}\theta \\ &\leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{(2\pi)^{\frac{d}{2}}\sqrt{\vartheta^2}}\int \exp\left(-\frac{1}{2}\left[\frac{1}{\vartheta^2} - \frac{\lambda^2 c_x^2}{2n}\right]\|\theta\|^2\right)\mathrm{d}\theta\end{aligned}$$

The integral is a properly defined Gaussian integral under the hypothesis that $\frac{1}{\vartheta^2} - \frac{\lambda^2 c_x^2}{2n} > 0$ hence $\lambda < \frac{1}{c_x}\sqrt{\frac{n}{\vartheta}^2}$. The integral is proportional to a Gaussian and we can directly write:

$$\pi\left\{\mathbb{E}\left[\exp\left(\lambda(R^H - r_n^H)\right)\right]\right\} \leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{\sqrt{1 - \frac{\vartheta^2\lambda^2 c_x^2}{2n}}}$$

writing everything in the exponential gives the desired result. □

Proof of Corollary 5.6.2. We apply Theorem 5.4.2 to get:

$$\begin{aligned}\mathcal{B}_\lambda(\mathcal{F}_1) &= \inf_{(\mathbf{m},\sigma^2)}\left\{\int R^H \mathrm{d}\Phi_{\mathbf{m},\sigma^2} + \frac{\lambda}{2n} - \frac{1}{\lambda}\log\left(1 - \frac{\vartheta^2\lambda^2 c_x^2}{2n}\right) + 2\frac{\mathcal{K}(\Phi_{\mathbf{m},\sigma^2}, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda}\right\} \\ &= \inf_{(m,\sigma^2)}\left\{\int R^H \mathrm{d}\Phi_{\mathbf{m},\sigma^2} + 2\frac{\sum_{j=1}^{d}\left[\frac{1}{2}\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2}\right] + \frac{\|m\|^2}{\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda}\right\} +\end{aligned}$$

$$\frac{\lambda}{2n} - \frac{1}{\lambda}\log\left(1 - \frac{\vartheta\lambda^2 c_x^2}{2n}\right) \quad .$$

We use the fact that the hinge loss is Lipschitz and that the (X_i) are uniformly bounded $\|X\|_\infty < c_x$. We get $R^H(\theta) \leq \bar{R}^H + c_x\sqrt{d}\|\theta - \bar{\theta}\|$ and restrict the infimum to distributions ν such that $m = \bar{\theta}$:

$$\mathcal{B}(\mathcal{F}_1) \leq \overline{R}^H + \inf_{\sigma^2}\left\{c_x d\sigma^2 + \frac{\lambda}{2n} - \frac{1}{\lambda}\log\left(1 - \frac{\vartheta^2\lambda^2 c_x^2}{2n}\right) + \frac{d\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{2d\sigma^2}{\vartheta^2} + \frac{2}{\vartheta^2} - d + 2\log\left(\frac{2}{\varepsilon}\right)}{\lambda}\right\}.$$

We specify $\sigma^2 = \frac{1}{\sqrt{dn}}$ and $\lambda = \frac{1}{c_x}\sqrt{\frac{n}{\vartheta^2}}$ such that we get:

$$\mathcal{B}(\mathcal{F}_1) \leq \overline{R^H} + c_x\sqrt{\frac{d}{n}} + \frac{\sqrt{\vartheta^2}}{2c_x\sqrt{n}} - c_x\sqrt{\frac{\vartheta^2}{n}}\log\left(1 - \frac{1}{2}\right) + d\frac{c_x\vartheta}{\sqrt{n}}\log\left(\vartheta^2\sqrt{nd}\right) + c_x\vartheta\frac{\frac{2d}{n\vartheta^2} + \frac{2}{\vartheta^2} - d + 2\log\left(\frac{2}{\varepsilon}\right)}{\sqrt{n}}.$$

To get the correct rate we take the prior variance to be $\vartheta^2 = \frac{1}{d}$ by replacing in the above equation we get the desired result.

□

Proof of Theorem 5.6.3. From Nesterov [2004] (th. 3.2.2) we have the following bound on the objective function minimized by VB, (the objective is not uniformly Lipschitz)

$$\rho^k(r_n^H) + \frac{1}{\lambda}\mathcal{K}(\rho^k, \pi) - \inf_{\rho\in\mathcal{F}_1}\left\{\rho(r_n^H) + \frac{1}{\lambda}\mathcal{K}(\rho, \pi)\right\} \leq \frac{LM}{\sqrt{1+k}}. \tag{5.17}$$

where the initial point was taken in a ball of radius M around $\tilde{\rho}$.

We have from equation (5.11) specified for measures ρ^k probability $1 - \varepsilon$,

$$\lambda\int R^H \mathrm{d}\rho^k \leq \lambda\int r_n^H \mathrm{d}\rho^k + f(\lambda, n) + \mathcal{K}(\rho^k, \pi) + \log\left(\frac{1}{\varepsilon}\right)$$

Combining the two equations yields,

$$\int R^H \mathrm{d}\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{1}{\lambda}f(n, \lambda) + \inf_{\rho\in\mathcal{F}_1}\left\{\rho(r_n^H) + \frac{1}{\lambda}\mathcal{K}(\rho, \pi)\right\} + \frac{1}{\lambda}\log\frac{1}{\varepsilon}$$

We can therefore write for any $\rho \in \mathcal{F}_1$,

$$\int R^H \mathrm{d}\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{1}{\lambda}f(n, \lambda) + \rho(r_n^H) + \frac{1}{\lambda}\mathcal{K}(\rho, \pi) + \frac{1}{\lambda}\log\frac{1}{\varepsilon}$$

Using equation (5.11) a second time we get with probability $1-\varepsilon$

$$\int R^H \mathrm{d}\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{2}{\lambda} f(n,\lambda) + \rho(R^H) + \frac{2}{\lambda}\mathcal{K}(\rho,\pi) + \frac{2}{\lambda}\log\frac{2}{\varepsilon}$$

Because this is true for any $\rho \in \mathcal{F}_1$ in $1-\varepsilon$ we can write the bound for the smallest measure in $\mathcal{F}_1$.

$$\int R^H \mathrm{d}\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{2}{\lambda} f(n,\lambda) + \inf_{\rho\in\mathcal{F}_1}\left\{\rho(R^H) + \frac{2}{\lambda}\mathcal{K}(\rho,\pi)\right\} + \frac{2}{\lambda}\log\frac{2}{\varepsilon}$$

By taking the Gaussian measure with variance $\frac{1}{dn}$ and mean $\overline{\theta}$ in the infimum and taking $\lambda = \frac{1}{c_x}\sqrt{nd}$ and $\vartheta = \frac{1}{d}$, we can plug in the result of Corrolary 5.6.2 to get the result.□

5.A.6 Proofs of Section 5.7

Proof of Lemma 5.7.1. The idea of the proof is to use Hoeffding's decomposition of U-statistics combined with Hoeffding's inequality for iid random variables. This was done in ranking by Clémençon et al. [2008a], and later in Ridgway [2015]; Robbiano [2013] for ranking via aggregation and Bayesian statistics. The proof is as follows: we define

$$q^\theta_{i,j} = \mathbf{1}_{(Y_i - Y_j)(f_\theta(X_i) - f_\theta(X_j))<0} - R(\theta)$$

so that

$$U_n := \frac{1}{n(n-1)}\sum_{i,j} q^\theta_{i,j} = r_n(\theta) - R(\Theta).$$

From Hoeffding [1948] we have

$$U_n = \frac{1}{n!}\sum_\pi \frac{1}{\lfloor \frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^\theta_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}$$

where the sum is taken over all the permutations π of $\{1,\dots,n\}$. Jensen's inequality leads to

$$\begin{aligned}\mathbb{E}\exp[\lambda U_n] &= \mathbb{E}\exp\left[\lambda\frac{1}{n!}\sum_\pi \frac{1}{\lfloor \frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^\theta_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right] \\ &\leq \frac{1}{n!}\sum_\pi \mathbb{E}\exp\left[\frac{\lambda}{\lfloor \frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^\theta_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right].\end{aligned}$$

We now use, for each of the terms in the sum the same argument as in the proof of Lemma 5.5.1 to get

$$\mathbb{E}\exp[\lambda U_n] \leq \frac{1}{n!}\sum_{\pi}\exp\left[\frac{\lambda^2}{2\lfloor\frac{n}{2}\rfloor}\right] \leq \exp\left[\frac{\lambda^2}{n-1}\right]$$

(in the last step, we used $\lfloor\frac{n}{2}\rfloor \geq (n-1)/2$). We proceed in the same way to upper bound $\mathbb{E}\exp[-\lambda U_n]$. $\square$

Proof of Lemma 5.7.2. As already done above, we use Bernstein inequality and Hoeffding decomposition. Fix θ. We define this time

$$q^{\theta}_{i,j} = \mathbf{1}\{\langle\theta, X_i - X_j\rangle\,(Y_i - Y_j) < 0\} - \mathbf{1}\{[\sigma(X_i) - \sigma(X_j)](Y_i - Y_j) < 0\} - R(\theta) + \overline{R}$$

so that

$$U_n := r_n(\theta) - \overline{r}_n - R(\theta) + \overline{R} = \frac{1}{n(n-1)}\sum_{i\neq j} q^{\theta}_{i,j}.$$

Then,

$$U_n = \frac{1}{n!}\sum_{\pi}\frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^{\theta}_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}.$$

Jensen's inequality:

$$\begin{aligned}\mathbb{E}\exp[\lambda U_n] &= \mathbb{E}\exp\left[\lambda\frac{1}{n!}\sum_{\pi}\frac{1}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^{\theta}_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right]\\ &\leq \frac{1}{n!}\sum_{\pi}\mathbb{E}\exp\left[\frac{\lambda}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^{\theta}_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right].\end{aligned}$$

Then, for each of the terms in the sum, use Bernstein's inequality:

$$\mathbb{E}\exp\left[\frac{\lambda}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^{\theta}_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right] \leq \exp\left[\frac{\mathbb{E}((q^{\theta}_{\pi(1),\pi(1+\lfloor\frac{n}{2}\rfloor)})^2)\frac{\lambda^2}{\lfloor\frac{n}{2}\rfloor}}{2\left(1-2\frac{\lambda}{\lfloor\frac{n}{2}\rfloor}\right)}\right].$$

We use again $\lfloor\frac{n}{2}\rfloor \geq (n-1)/2$. Then, as the pairs (X_i, Y_i) are iid, we have $\mathbb{E}((q^{\theta}_{\pi(1),\pi(1+\lfloor\frac{n}{2}\rfloor)})^2) = \mathbb{E}((q^{\theta}_{1,2})^2)$ and then $\mathbb{E}((q^{\theta}_{1,2})^2) \leq C[R(\theta) - \overline{R}]$ thanks to the margin assumption. So

$$\mathbb{E}\exp\left[\frac{\lambda}{\lfloor\frac{n}{2}\rfloor}\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor} q^{\theta}_{\pi(i),\pi(i+\lfloor\frac{n}{2}\rfloor)}\right] \leq \exp\left[\frac{C[R(\theta) - \overline{R}]\frac{\lambda^2}{n-1}}{\left(1-\frac{4\lambda}{n-1}\right)}\right].$$

This ends the proof of the proposition. □
Proof of Corollary 5.7.4. The calculations are similar to the ones in the proof of Corollary 5.5.4 so we don't give the details. Note that when we reach

$$\mathcal{B}_\lambda(\mathcal{F}_1) \leq \overline{R} + \frac{2\lambda}{n-1} + \frac{c\sqrt{d} + d\log(2\lambda) + 2\log\left(\frac{2\mathrm{e}}{\varepsilon}\right)}{\lambda},$$

an approximate minimization with respect to λ leads to the choice $\lambda = \sqrt{\frac{d(n-1)}{2}}$. □

5.A.7 Proofs of Section 5.8

Proof. First, note that, for any ρ,

$$\begin{aligned}\mathcal{K}(\rho, \pi_\beta) &= \beta \int (R - \overline{R}) \mathrm{d}\rho + \mathcal{K}(\rho, \pi) + \log \int \exp\left[-\beta(R - \overline{R})\right] \mathrm{d}\pi \\ &\leq \beta \int (R - \overline{R}) \mathrm{d}\rho + \mathcal{K}(\rho, \pi).\end{aligned}$$

Now, we define a subset of $\mathcal{F}$ that will be used for the calculation of the bound. We define for $\delta > 0$ the probability distribution $\rho_{U,V,\delta}(\mathrm{d}\theta)$ as π conditioned to $\theta = \mu\nu^T$ with μ is uniform on $\{\forall(i,\ell), |\mu_{i,\ell} - U_{i,\ell}| \leq \delta\}$ and ν is uniform on $\{\forall(j,\ell), |\nu_{i,\ell} - V_{j,\ell}| \leq \delta\}$. Note that

$$\begin{aligned}\int (R - \overline{R}) \mathrm{d}\rho_{M,N,\delta} &= \int \mathbb{E}((\theta_X - M_X)^2) \rho_{U,V,\delta}(\mathrm{d}\theta) \\ &\leq \int 3\mathbb{E}(((UV^T)_X - M_X)^2) \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \\ &\quad + 3\int \mathbb{E}(((U\nu^T)_X - (UV^T)_X)^2) \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \\ &\quad + 3\int \mathbb{E}(((\mu\nu^T)_X - (U\nu^T)_X)^2) \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)).\end{aligned}$$

By definition, the first term is $= 0$. Moreover:

$$\begin{aligned}&\int \mathbb{E}(((U\nu^T)_X - (UV^T)_X)^2) \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \\ &= \int \frac{1}{m_1 m_2} \sum_{i,j} \left[\sum_k U_{i,k}(\nu_{j,k} - V_{j,k})\right]^2 \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \\ &\leq \int \frac{1}{m_1 m_2} \sum_{i,j} \left[\sum_k U_{i,k}^2\right] \left[\sum_k (\nu_{j,k} - V_{j,k})^2\right] \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu))\end{aligned}$$

$$\leq KrC^2\delta^2.$$

In the same way,

$$\int \mathbb{E}(((\mu\nu^T)_X - (U\nu^T)_X)^2)\rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \leq \int \|\mu - U\|_F^2 \|\nu\|_F^2 \rho_{U,V,\delta}(\mathrm{d}(\mu,\nu)) \leq Kr(C+\delta)^2\delta^2.$$

So:

$$\int (R - \overline{R})\mathrm{d}\rho_{M,N,\delta} \leq 2Kr\delta^2(C+\delta^2).$$

Now, let us consider the term $\mathcal{K}(\rho_{U,V,\delta}, \pi)$. An explicit calculation is possible but tedious. Instead, we might just introduce the set $\mathcal{G}_\delta = \{\theta = \mu\nu^T, \|\mu - U\|_F \leq \delta, \|\nu - V\|_F \leq \delta\}$ and note that $\mathcal{K}(\rho_{U,V,\delta}, \pi) \leq \log \frac{1}{\pi(\mathcal{G}_\delta)}$. An upper bound for $\mathcal{G}_\delta$ is calculated page 317-320 in Alquier [2014] and the result is given by (10) in this reference:

$$\mathcal{K}(\rho_{U,V,\delta}, \pi) \leq 4\delta^2 + 2\|U\|_F^2 + 2\|N\|_F^2 + 2\log(2) + (m_1+m_2)r\log\left(\frac{1}{\delta}\sqrt{\frac{3\pi(m_1 \vee m_2)K}{4}}\right) + 2K\log\left(\frac{\Gamma(a)3^{a+1}\exp(2)}{b^{a+1}2^a}\right)$$

as soon as the restriction $b \leq \frac{\delta^2}{2m_1K\log(2m_1K)}, \frac{\delta^2}{2m_2K\log(2m_2K)}$ is satisfied. So we obtain:

$$\mathcal{K}(\rho_{U,V,\delta}, \pi_\beta) \leq \beta 2Kr\delta^2(C+\delta^2) + 4\delta^2 + 2\|U\|_F^2 + 2\|N\|_F^2 + 2\log(2) + (m_1+m_2)r\log\left(\frac{1}{\delta}\sqrt{\frac{3\pi(m_1 \vee m_2)K}{4}}\right) + 2K\log\left(\frac{\Gamma(a)3^{a+1}\exp(2)}{b^{a+1}2^a}\right).$$

Note that $\|U\|_F^2 \leq C^2 r m_1$, $\|V\|_F^2 \leq C^2 r m_2$ and $K \leq m_1 + m_2$ so it is clear that the choice $\delta = \sqrt{\frac{1}{\beta}}$ and $b \leq \frac{1}{2\beta(m_1 \vee m_2)\log(2K(m_1 \vee m_2))}$ leads to the existence of a constant $\mathcal{C}(a, C)$ such that

$$\mathcal{K}(\rho_{U,V,\delta}, \pi_\beta) \leq \mathcal{C}(a,C)\left\{r(m_1+m_2)\log\left[\beta b(m_1+m_2)K\right] + \frac{1}{\beta}\right\}.$$

□

5.B Implementation details

5.B.1 Sequential Monte Carlo

Tempering SMC approximates iteratively a sequence of distribution ρ_{λ_t}, with

$$\rho_{\lambda_t}(\mathrm{d}\theta) = \frac{1}{Z_t}\exp\left(-\lambda_t r_n(\theta)\right)\pi(\mathrm{d}\theta),$$

and temperature ladder $\lambda_0 = 0 < \ldots < \lambda_T = \lambda$. The pseudo code below is given for an adaptive sequence of temperatures.

Algorithm 19 Tempering SMC

Input N (number of particles), $\tau \in (0,1)$ (ESS threshold), $\kappa > 0$ (random walk tuning parameter)

Init. Sample $\theta_0^i \sim \pi_\xi(\theta)$ for $i = 1$ to N, set $t \leftarrow 1$, $\lambda_0 = 0$, $Z_0 = 1$.

Loop a. Solve in λ_t the equation

$$\frac{\{\sum_{i=1}^N w_t(\theta_{t-1}^i)\}^2}{\sum_{i=1}^N \{w_t(\theta_{t-1}^i))^2\}} = \tau N, \quad w_t(\theta) = \exp[-(\lambda_t - \lambda_{t-1}) r_n(\theta)] \tag{5.18}$$

using bisection search. If $\lambda_t \geq \lambda_T$, set $Z_T = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$, and stop.

b. Resample: for $i = 1$ to N, draw A_t^i in $1, \ldots, N$ so that $\mathbb{P}(A_t^i = j) = w_t(\theta_{t-1}^j) / \sum_{k=1}^N w_t(\theta_{t-1}^k)$; see Algorithm 20 in the appendix.

c. Sample $\theta_t^i \sim M_t(\theta_{t-1}^{A_t^i}, \mathrm{d}\theta)$ for $i = 1$ to N where M_t is a MCMC kernel that leaves invariant π_t; see comments below.

d. Set $Z_t = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$.

The algorithm outputs a weighted sample (w_T^i, θ_T^i) approximately distributed as target posterior, and an unbiased estimator of the normalizing constant Z_{λ_T}.

Step **b.** of algorithm 5.B.1 depends of a resampling algorithm. We choose to use Systematic resampling, described in Algorithm 20.

Algorithm 20 Systematic resampling

Input: Normalised weights $W_t^j := w_t(\theta_{t-1}^j) / \sum_{i=1}^N w_t(\theta_{t-1}^i)$.

Output: indices $A^i \in \{1, \ldots, N\}$, for $i = 1, \ldots, N$.

a. Sample $U \sim \mathcal{U}([0, 1])$.

b. Compute cumulative weights as $C^n = \sum_{m=1}^n NW^m$.

c. Set $s \leftarrow U$, $m \leftarrow 1$.

d. **For** $n = 1 : N$

While $C^m < s$ **do** $m \leftarrow m + 1$.

$A^n \leftarrow m$, and $s \leftarrow s + 1$.

End For

For the MCMC step, we used a Gaussian random-walk Metropolis kernel, with a covariance matrix for the random step that is proportional to the empirical covariance matrix of the current set of simulations.

5.B.2 Optimizing the bound

A natural idea to find a global optimum of the objective is to try to solve a sequence of local optimization problems with increasing inverse temperatures. For inverse temperature $\lambda = 0$ the problem can be solved exactly (as a KL divergence between two Gaussians). Then, for two consecutive temperatures, the corresponding solutions should be close enough.

This idea has been coined under several names. It has a long history in variational Bayes litterature under the name deterministic annealing. Yuille uses it on mean field on Gibbs distribution for Markov random fields. In addition the intermediate results can be of interest in our case for selecting the temperature. One can compute the bound at almost no additional cost as a function of the current risk. In turns this can be used to monitor the bound.

Algorithm 21 Deterministic annealing

Input $(\lambda_t)_{t\in[0,T]}$ a sequence of inverse temperature

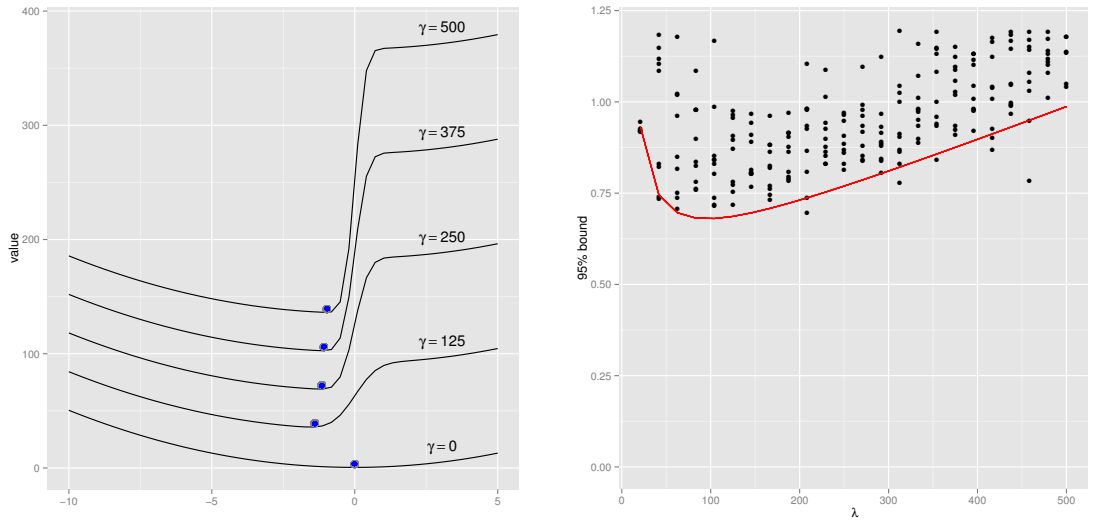
Init. Set $m = 0$ and $\Sigma = \vartheta I_d$, the values minimizing the KL-divergence for $\lambda = 0$

Loop t=1,...,T

a. $m^{\lambda_t}, \Sigma^{\lambda_t}$ = Minimize $\mathcal{L}^{\lambda_t}(m, \Sigma)$ using some local optimization routine with initial points $m^{\lambda_{t-1}}, \Sigma^{\lambda_{t-1}}$

b. Break if the empirical bound increases.

End Loop



(a) A one dimensional problem

(b) Empirical bound

Figure 5.B.1: Deterministic annealing on a Pima Indians with one covariate and full model resp.

The right panel gives the empirical bound obtained for the DA method (in red) and the dot are direct global optimization based on L-BFGS algorithms from starting values drawn from the prior. Each optimization problem is repeated 20 times.

We find that using a deterministic annealing algorithm with a limited amount of steps helps in finding a high enough optimum. On the left panel of Figure 5.B.1,

we can see the one dimensional case where the initial problem $\gamma = 0$ corresponds to a convex minimization problem and where the increasing temperature gradually complexifies the optimization problem. Figure 5.B.1 shows that the solution given by DA is in average lower than randomly initialized optimization.

5.C Stochastic gradient descent

The stochastic gradient descent algorithm used in Section 5.7 is described as Algorithm 22.

Algorithm 22 Stochastic Gradient Descent

Input B a batch size, an unbiased estimator of the gradient $\hat{\nabla}_B f$, $\eta \in (0, 1)$ and c

While $\neg$converged

a. $x_{t+1} = x_t - \lambda_t \hat{\nabla}_B f(x_t)$

b. Update $\lambda_{t+1} = \frac{1}{(t+c)^\eta}$

End Loop

In all our experiment we take $c = 1$ and $\eta = 0.9$.

6

Towards the automatic calibration of the number of particles in SMC^2

This is joint work with Nicolas Chopin, Mathieu Gerber and Omiros Papaspiliopoulos
Status: To be presented at *SYSID 2015*

6.1 Introduction

Consider a state-space model, with parameter $\theta \in \Theta$, latent Markov process $(x_t)_{t\geq 0}$, and observed process $(y_t)_{t\geq 0}$, taking values respectively in $\mathcal{X}$ and $\mathcal{Y}$. The model is defined through the following probability densities: θ has prior $p(\theta)$, $(x_t)_{t\geq 0}$ has initial law $\mu_\theta(x_0)$ and Markov transition $f_\theta^X(x_t|x_{t-1})$, and the y_t's are conditionally independent, given the x_t's, with density $f_\theta^Y(y_t|x_t)$. Sequential analysis of such a model amounts to computing recursively (in t) the posterior distributions

$$p(\theta, x_{0:t}|y_{0:t}) = \frac{p(\theta)\mu_\theta(x_0)}{p(y_{0:t})} \left\{ \prod_{s=1}^{t} f_\theta^X(x_s|x_{s-1}) \right\} \left\{ \prod_{s=0}^{t} f_\theta^Y(y_s|x_s) \right\}$$

or some of its marginals (e.g. $p(\theta|y_{0:t})$); the normalising constant $p(y_{0:t})$ of the above density is the marginal likelihood (evidence) of the data observed up to time t.

For a fixed θ, the standard approach to sequential analysis of state-space models is particle filtering: one propagates N_x particles in $\mathcal{X}$ over time through mutation steps (based on proposal distribution $q_{t,\theta}(x_t|x_{t-1})$ at time t) and resampling steps; see Algorithm 23. Note the conventions: $1 : N_x$ denotes the set of integers

$\{1, \ldots, N_x\}$, $y_{0:t}$ is $(y_0, \ldots, y_t)$, $x_t^{1:N_x} = (x_t^1, \ldots, x_t^{N_x})$, $x_{0:t}^{1:N_x} = (x_0^{1:N_x}, \ldots, x_t^{1:N_x})$, and so on.

Algorithm 23 Particle filter (PF, for fixed θ)

Operations involving superscript n must be performed for all $n \in 1 : N_x$.
At time 0:

(a) Sample $x_0^n \sim q_{0,\theta}(x_0)$.

(b) Compute weights

$$w_{0,\theta}(x_0^n) = \frac{\mu_\theta(x_0^n) f^Y(y_0|x_0^n)}{q_{0,\theta}(x_0^n)}$$

normalised weights, $W_{0,\theta}^n = w_{0,\theta}(x_0^n) / \sum_{i=1}^{N_x} w_{0,\theta}(x_0^i)$, and incremental likelihood estimate
$\hat{\ell}_0(\theta) = N_x^{-1} \sum_{n=1}^{N_x} w_{0,\theta}^n$.

Recursively, from time $t = 1$ to time $t = T$:

(a) Sample $a_t^n \sim \mathcal{M}(W_{t-1,\theta}^{1:N_x})$, the multinomial distribution which generates value $i \in 1 : N_x$ with probability $W_{t-1,\theta}^i$.

(b) Sample $x_t^n \sim q_{t,\theta}(\cdot|x_{t-1}^{a_t^n})$.

(c) Compute weights

$$w_{t,\theta}(x_{t-1}^{a_t^n}, x_t^n) = \frac{f^X(x_t^n|x_{t-1}^{a_t^n}) f^Y(y_t|x_t^n)}{q_{t,\theta}(x_t^n|x_{t-1}^{a_t^n})}$$

$$W_{t,\theta}^n = \frac{w_{t,\theta}(x_{t-1}^{a_t^n}, x_t^n)}{\sum_{i=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_t^i}, x_t^i)}$$

and incremental likelihood estimate
$\hat{\ell}_t(\theta) = N_x^{-1} \sum_{n=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_t^n}, x_t^n)$.

The output of Algorithm 23 may be used in different ways: at time t, the quantity $\sum_{n=1}^{N_x} W_{t,\theta}^n \varphi(x_t^n)$ is a consistent (as $N_x \to +\infty$) estimator of the filtering expectation $\mathbb{E}[\varphi(x_t)|y_{0:t}, \theta]$; In addition, $\hat{\ell}_t(\theta)$ is an *unbiased* estimator of incremental likelihood $p(y_t|y_{0:t-1}, \theta)$, and $\prod_{s=0}^{t} \hat{\ell}(\theta)$ is an unbiased estimator of the full likelihood $p(y_{0:t}|\theta)$ [Del Moral, 1996b, Lemma 3].

In order to perform joint inference on parameter θ and state variables, Chopin et al.

[2013a] derived the SMC^2 sampler, that is, a SMC (Sequential Monte Carlo) algorithm in $\theta-$space, which generates and propagates N_θ values θ^m in Θ, and which, for each θ^m, runs a particle filter (i.e. Algorithm 23) for $\theta = \theta^m$, of size N_x. One issue however is how to choose N_x: if too big, then CPU time is wasted, while if taken too small, then the performance of the algorithm deteriorates. Chopin et al. [2013a] give formal results (adapted from Andrieu et al. [2010b]) that suggest that N_x should grow at a linear rate during the course of the algorithm. They also propose a practical method for increasing N_x adaptively, based on an importance sampling step where the N_θ particle systems, of size N_x, are replaced by new particle systems of size N_x^{new}. But this importance sampling step increases the degeneracy of the weights, which in return may lead to more frequent resampling steps, which are expensive. In this chapter, we derive an alternative way to increase N_x adaptively, which is not based on importance sampling, but rather on a CSMC (conditional Sequential Monte Carlo) update, which is less CPU intensive.

6.2 Background on SMC^2

6.2.1 IBIS

To explain SMC^2, we first recall the structure of the IBIS algorithm [Chopin, 2002b] as Algorithm 24. For a model with parameter $\theta \in \Theta$, prior $p(\theta)$, data $y_{0:T}$, and incremental likelihood $p(y_t|y_{0:t-1}, \theta)$, IBIS provides at each iteration t an approximation of partial posterior $p(\theta|y_{0:t})$. In practice, IBIS samples N_θ particles θ^m from the prior, then perfoms sequential importance sampling steps, from $p(\theta|y_{0:t-1})$ to $p(\theta|y_{0:t})$ using incremental weight $p(\theta|y_{0:t})/p(\theta|y_{0:t-1}) \propto p(y_t|y_{0:t-1}, \theta)$.

To avoid weight degeneracy, one performs a resample-move step (described as Step (b) in Algorithm 24). When the ESS (effective sample size) of the weights, computed as:

$$\text{ESS}(\omega^{1:N_\theta}) = \frac{(\sum_{m=1}^{N_\theta} \omega^m)^2}{\sum_{m=1}^{N_\theta} (\omega^m)^2} \in [1, N]$$

goes below some threshold $\text{ESS}_{\min}$ (e.g. $N/2$), the θ^m's are resampled, then moved according to some Markov kernel K_t that leaves invariant the current target of the algorithm, $p(\theta|y_{0:t})$. This resample-move step re-introduces diversity among the θ-particles.

A convenient default choice for K_t is several iterations of random-walk Metropolis, with the random step calibrated to the spread of the current particle population (i.e. variance of random step equals some fraction of the covariance matrix of the resampled particles).

The main limitation of IBIS is that it requires evaluating the likelihood increment $p(y_t|y_{0:t-1}, \theta)$, which is typically intractable for state-space models. On the

Algorithm 24 IBIS

Operations involving superscript m must be performed for all $m \in 1 : N_\theta$.

(Init) Sample $\theta^m \sim p(\theta)$, set $\omega^m \leftarrow 1$.

From time $t = 0$ to time $t = T$, do

(a) Update importance weights

$$\omega^m \leftarrow \omega^m \times p(y_t|y_{0:t-1}, \theta).$$

(b) If $\mathrm{ESS}(\omega^{1:N_\theta}) \leq \mathrm{ESS}_{\min}$, sample (for all m) $\tilde{\theta}^m$ from mixture

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t(\theta^m, d\theta),$$

where K_t is a Markov kernel with invariant distribution $p(\theta|y_{0:t})$; finally reset particle system to

$$\theta^{1:N_\theta} \leftarrow \tilde{\theta}^{1:N_\theta}, \quad \omega^{1:N_\theta} \leftarrow (1, \ldots, 1).$$

other hand, we have seen that this quantity may be estimated unbiasedly by particle filtering. This suggests combining IBIS (i.e. SMC in the θ-dimension) with particle filtering (i.e. SMC in the x_t-dimension), as done in the SMC^2 algorithm.

6.2.2 SMC^2

The general structure of SMC^2 is recalled as Algorithm 25. Essentially, one recognises the IBIS algorithm, where the intractable incremental weight $p(y_t|y_{0:t-1}, \theta^m)$ has been replaced by the unbiased estimate $\hat{\ell}_t(\theta^m)$. This estimate is obtained from a PF run for $\theta = \theta^m$; thus N_θ PFs are run in parallel. Denote $(x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$ the random variables generated by the PF associated to θ^m.

This 'double-layer' structure suggests that SMC^2 suffers from two levels of approximation, and as such that it requires both $N_x \to +\infty$ and $N_\theta \to +\infty$ to converge. It turns out however that SMC^2 is valid for any fixed value of N_x; that is, for any fixed $N_x \geq 1$, it converges as $N_\theta \to +\infty$.

This property is intuitive in the simplified case when resampling-move steps are never triggered (i.e. take $\mathrm{ESS}_{\min} = 0$). Then SMC^2 collapses to importance

Algorithm 25 SMC^2

Operations involving superscript m must be performed for all $m \in 1 : N_\theta$.

(Init) Sample $\theta^m \sim p(\theta)$, set $\omega^m \leftarrow 1$.

From time $t = 0$ to time $t = T$, do

(a) For each θ^m, run iteration t of Algorithm 23, so as to obtain $(x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$, and $\hat{\ell}_t(\theta^m)$.

(b) Update weights

$$\omega^m \leftarrow \omega^m \times \hat{\ell}_t(\theta^m).$$

(c) If $\mathrm{ESS}(\omega^{1:N_\theta}) \leq \mathrm{ESS}_{\min}$, sample (for all m) $(\tilde{\theta}^m, \tilde{x}_{0:t}^{1:N_x,m}, \tilde{a}_{1:t}^{1:N_x,m})$ from mixture

$$\frac{1}{\sum_{m=1}^{N_\theta} \omega^m} \sum_{m=1}^{N_\theta} \omega^m K_t \left((\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m}), d\cdot \right),$$

where K_t is a PMCMC kernel with invariant distribution $\pi_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$ (see text); finally reset particle system to

$$(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m}) \leftarrow (\tilde{\theta}^m, \tilde{x}_{0:t}^{1:N_x,m}, \tilde{a}_{1:t}^{1:N_x,m})$$

and $\omega^m \leftarrow 1$, for all m.

sampling, with weights replaced by unbiased estimates, and it is easy to show convergence from first principles.

We now give a brief outline of the formal justification of SMC^2 for fixed N_x, and refer to Chopin et al. [2013a] for more details. SMC^2 may be formalised as a SMC sampler for the sequence of extended distributions:

$$\pi_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x}) = \frac{p(\theta)}{p(y_{0:t})} \psi_{t,\theta}(x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x}) \prod_{s=0}^{t} \hat{\ell}_s(\theta)$$

where $\psi_{t,\theta}$ denotes the joint pdf of the random variables generated by a PF up to time t (for parameter θ), and $\hat{\ell}_s(\theta)$ denotes the unbiased estimate of the likelihood increment computed from that PF, $\hat{\ell}_0(\theta) = N_x^{-1} \sum_{n=1}^{N} w_0(x_0^n)$, $\hat{\ell}_s(\theta) = N_x^{-1} \sum_{n=1}^{N} w_{s,\theta}(x_{s-1}^{a_t^n}, x_s^n)$ for $s > 0$; i.e. $\hat{\ell}_s(\theta)$ is actually a function of $(\theta, x_{0:s}^{1:N_x}, a_{1:s}^{1:N_x})$.

One recognises in π_t the type of extended target distribution simulated by PMCMC (Particle MCMC, Andrieu et al. [2010b]) algorithms. Note π_t is a proper

probability density (it integrates to one), and that the marginal distribution of θ is $p(\theta|y_{0:t})$. These two properties are easily deduced from the unbiasedness of $\prod_{s=0}^{t} \hat{\ell}_s(\theta)$ (as an estimator of $p(y_{0:t}|\theta)$). In addition,

$$\pi_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x}) = \pi_{t-1}(\theta, x_{0:t-1}^{1:N_x}, a_{1:t-1}^{1:N_x}) \frac{\psi_{t,\theta}(x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})}{\psi_{t-1,\theta}(x_{0:t-1}^{1:N_x}, a_{1:t-1}^{1:N_x})} \hat{\ell}_t(\theta)$$

where one recognises in the second factor the distribution of the variables generated by a PF at time t, conditional on those variables generated up to time $t-1$. Thus, the equation above justifies both Step (a) of Algorithm 25, where the particle filters are extended from time $t-1$ to t, and Step (b), where the particles $(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$ are reweighted by $\hat{\ell}_t(\theta^m)$.

We describe in the following section PMCMC moves that may be used in Step (c). Before, we note that a naive implementation of SMC^2 has a $\mathcal{O}(tN_xN_\theta)$ memory cost at time t, as one must stores in memory $(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$ for each $m \in 1 : N_\theta$. This memory cost may be substantial even on a modern computer.

6.2.3 PMCMC moves

To make more explicit the dependence of the unbiased estimate of the likelihood on the variables generated during the course of PF, define

$$L_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x}) = \prod_{s=0}^{t} \hat{\ell}_s(\theta) = \left\{ \frac{1}{N_x} \sum_{n=1}^{N_x} w_{0,\theta}(x_0^n) \right\} \prod_{s=1}^{t} \left\{ \frac{1}{N_x} \sum_{n=1}^{N_x} w_{s,\theta}(x_{s-1}^{a_s^n}, x_s^n) \right\}.$$

The PMMH (Particle Markov Metropolis-Hastings) kernel, described as Algorithm 26, may be described informally as a Metropolis step in θ-space, where the likelihood of both the current value and the proposed value have been replaced by unbiased estimators. Formally, as proven in Andrieu et al. [2010b], it is in fact a standard Metropolis step with respect to the extended distribution $\pi_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$; in particular it leaves invariant $p(\theta|y_{0:t})$. (For convenience, our description of PMMH assumes a random walk proposal, but PMMH is not restricted to this kind of proposal.)

In practice, we set Σ_t, the covariance matrix of the proposal, to a fraction of the covariance matrix of the resampled θ-particles.

One advantage of using PMHMH within SMC^2 is that it does not require storing all the variables generated by the N_θ PFs: operations at time $t > 0$ require only having access to, for each m, $(\theta^m, x_{t-1}^{1:N_x,m}, a_{t-1}^{1:N_x,m})$ and $L_{t-1}(\theta^m, x_{0:t-1}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$, which is computed recursively. Memory cost then reduces to $\mathcal{O}(N_\theta N_x)$.

The Particle Gibbs approach is an alternative PMCMC step, based on the following property of target π_t: if one extends π_t with random index k, such that

Algorithm 26 Random walk PMMH update

Input: $(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$
Output: $(\tilde{\theta}, \tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x})$

1. $\theta^\star = \theta + z$, $z \sim N(0, \Sigma_t)$.

2. Generate PF (Algorithm 23) for parameter $\theta^\star$; let $(x_{0:t}^{1:N_x,\star}, a_{1:t}^{1:N_x,\star})$ the output.

3. With probability $1 \wedge r$,

$$r = \frac{p(\theta^\star) L_t(\theta^\star, x_{0:t}^{1:N_x,\star}, a_{1:t}^{1:N_x,\star})}{p(\theta) L_t(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})}$$

let $(\tilde{\theta}, \tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x}) \leftarrow (\theta^\star, x_{0:t}^{1:N_x,\star}, a_{1:t}^{1:N_x,\star})$; otherwise $(\tilde{\theta}, \tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x}) \leftarrow (\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$.

$k \in 1 : N_x$, and $k \sim \mathcal{M}(W_T^{1:N_x})$, the normalised weights at the final iteration, then (a) the selected trajectory, together with θ, follow the posterior distribution $p(\theta, x_{0:t}|y_{0:t})$; and (b) the remaining arguments of π_t follow a CSMC (conditional SMC) distribution, which corresponds to the distribution of the random variables generated by a PF, but conditional on one trajectory fixed to the selected trajectory; see Algorithm 27.

In contrast with PMMH, implementing particle Gibbs steps within SMC^2 requires having access to all the variables $(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$ at time t, which as we have already discussed, might incur too big a memory cost.

6.2.4 Choosing N_x

Andrieu et al. [2010b] show that, in order to obtain reasonable performance for PMMH, one should take $N_x = \mathcal{O}(t)$. Andrieu et al. [2013] show a similar result for Particle Gibbs.

In the context of SMC^2, this suggests that N_x should be allowed to increase in the course of the algorithm. To that effect, Chopin et al. [2013a] devised an exchange step, which consists in exchanging the current particle systems, of size N_x, with new particle systems, of size N_x^{new}, through importance sampling. In Chopin et al. [2013a]'s implementation, the exchange step is triggered each time the acceptance rate of the PMMH step (as performed in Step 3. of Algorithm 26) is below a certain threshold, and $N_x^{\text{new}} = 2N_x$ (i.e. N_x doubles every time).

The main drawback of this approach is that it introduces some weight degener-

Algorithm 27 Particle Gibbs update

Input: $(\theta, x_{0:t}^{1:N_x}, a_{1:t}^{1:N_x})$
Output: $(\tilde{\theta}, \tilde{x}_{0:t}^{1:N_x}, \tilde{a}_{1:t}^{1:N_x})$

1. Sample $b_t \sim \mathcal{M}(W_t^{1:N_x})$, with $W_t^n = w_{t,\theta}(x_{t-1}^{a_t^n}, x_t^n) / \sum_{i=1}^{N_x} w_{t,\theta}(x_{t-1}^{a_t^i}, x_t^i)$. From $s = t-1$ to $s = 0$, set $b_s \leftarrow a_{s+1}^{b_{s+1}}$. Set $\tilde{x}_s^1 \leftarrow x_s^{b_s}$, $\tilde{a}_s^1 = 1$ for all $s \in 0:T$.

2. Sample $\tilde{\theta}$ from a MCMC step that leaves invariant distribution $p(\theta | x_{0:t}, y_{0:t})$, but with $x_{0:t}$ set to $\tilde{x}_{0:t}^1$.

3. Sample $(\tilde{x}_{0:t}^{2:N_x}, \tilde{a}_{1:t}^{2:N_x})$ as in Algorithm 23, but for parameter $\tilde{\theta}$ and conditionally on $\tilde{x}_{0:t}^1$, that is: at time 0, generate $\tilde{x}_0^n \sim q_{0,\tilde{\theta}}$ for $n \in 2:N$, at time 1, sample $a_t^n \sim \mathcal{M}(W_1^{1:N_x})$, for $n \in 2:N$, and $x_t^n \sim q_{1,\tilde{\theta}}(\cdot | \tilde{x}_{t-1}^{\tilde{a}_1^n})$, and so on.

acy immediately after the resampling step. In particular, we will observe in our simulations that this prevents us from changing N_x too frequently, as the ESS of the weights then becomes too low.

In this chapter, we discuss how to use a Particle Gibbs step in order to increase N_x without changing the weights.

6.3 Proposed approach

6.3.1 Particle Gibbs and memory cost

We first remark that the Particle Gibbs step, Algorithm 27, offers a very simple way to change N_x during the course of the algorithm: In Step (2), simply regenerate a particle system (conditional on selected trajectory $\tilde{x}_{0:t}^1$) of size N_x^{new}. But, as already discussed, such a strategy requires then to access past particle values x_s^n (and also a_s^n), rather than only current particle values x_t^n.

This problem may be addressed in two ways. First, one may remark that, to implement Particle Gibbs, one needs to store only those x_s^n (and a_s^n) which have descendant among the N_x current particles x_t^n. Jacob et al. [2013] developed such a path storage approach, and gave conditions on the mixing of Markov chain (x_t) under which this approach has memory cost $\mathcal{O}(t + N_x \log N_x)$ (for a single PF with N_x particles, run until time t). Thus, an implementation of this approach within SMC^2 would lead to a $\mathcal{O}(N_\theta(t + N_x \log N_x))$ memory cost.

A second approach, developed here, exploits the deterministic nature of PRNGs (pseudo-random number generators): a sequence $z_0, z_1, \ldots, z_i, \ldots$ of computer-

generated random variates is actually a deterministic sequence determined by the initial state (seed) of the PRNG. It is sufficient to store that initial state and z_0 in order to recover any z_i in the future. The trade-off is an increase in CPU cost, as each access to z_i require re-computing $z_1, \ldots, z_i$.

We apply this idea to the variables $(x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$. By close inspection of Algorithm 25, we note that variables in a 'time slice' $(x_s^{1:N_x,m}, a_s^{1:N_x,m})$, $0 < s \leq t$ (or $x_0^{1:N_x,m}$ at time 0) are always generated jointly, either during Step (a), or during Step (c). In both cases, this time-slice is a deterministic function of the current PRNG state and the previous time slice. Thus, one may recover any time slice (when needed) by storing only (i) the PNRG state (immediately before the generation of the time slice); and (ii) in which Step (either (a) or (c)) the time slice was generated. This reduces the memory cost of SMC^2 from $\mathcal{O}(tN_\theta N_x)$ to $\mathcal{O}(N_\theta(t + N_x))$.

Compared to the path storage approach mentioned above, our PRNG recycling approach has a larger CPU cost, a smaller memory cost, and does not require any conditions on the mixing properties of process (x_t). Note that the CPU cost increase is within a factor of two, because each time a Particle Gibbs update is performed, the number of random variables that must be re-generated (i.e. the x_s^n and a_s^n in Algorithm 27) roughly equals the number of random variables that are generated for the first time (i.e. the $\tilde{x}_s^n$ and $\tilde{a}_s^n$ in Algorithm 27).

6.3.2 Nonparametric estimation of N_x

As seen in Algorithm 25, a Particle Gibbs step will be performed each time the ESS goes below some threshold. That the ESS is low may indicate that N_x is also too low, and therefore that the variance of the likelihood estimates $L_t(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$ is too high. Our strategy is to update (each time a Particle Gibbs step is performed) the current value of N_x to $N_x^{\text{new}} = \tau/\hat{\sigma}^2$, where $\hat{\sigma}^2$ is some (possibly rough) estimate of the variance of the *log* likelihood estimates. This is motivated by results from Doucet et al. [2012], who also develop some theory that supports choosing $\tau \approx 1$ is optimal (although their optimality results do not extend straightforwardly to our settings).

Assume $\Theta \subset \mathbb{R}^d$. To estimate σ^2, we use backfitting to fit a GAM (generalized additive model) to the responses $R^m = \log L_t(\theta^m, x_{0:t}^{1:N_x,m}, a_{1:t}^{1:N_x,m})$:

$$R^m = \alpha + \sum_{j=1}^{d} f_j(C_j^m) + \varepsilon^m,$$

using as covariates C_j^m the d principal components of the resampled θ-particles. The estimate σ^2 is then the empirical variance of the residuals. See e.g. Chap. 9 of Hastie et al. [2009] for more details on backfitting and GAM modelling.

We found this strategy to work well, with the caveat that choosing τ required some trial and error.

6.3.3 Additional considerations

Using Particle Gibbs as our PMCMC move within SMC^2 hast two advantages: (a) it makes it possible to change N_x without changing the weights, as explained above; and (b) it also makes it possible to update the θ^m according to Gibbs or Metropolis step that leaves $\theta|x_{0:t}, y_{0:t}$ invariant); see Step (3) of Algorithm 27. For models where sampling from $\theta|x_{0:t}, y_{0:t}$ is not convenient, one may instead update θ through several PMMH steps performed after the Particle Gibbs step.

6.4 Numerical example

We consider the following stochastic volatility model: $x_0 \sim N(\mu, \sigma^2/(1-\rho^2))$, $x_t - \mu = \rho(x_{t-1} - \mu) + \sigma\epsilon_t, \quad \epsilon_t \sim N(0,1)$ and $y_t|x_t \sim N(0, e^{x_t})$; thus $\theta = (\mu, \rho, \sigma)$, with $\rho \in [-1, 1]$, $\sigma > 0$. We assign independent priors to the components of θ: $\mu \sim N(0, 2^2)$, $\rho \sim N(0, 1)$ constrained to $[-1, 1]$, and $\sigma^2 \sim IG(3, 0.5)$. The dataset consists in log-returns from the monthly SP500 index, observed from 29/05/2013 to 19/12/2014; $T = 401$.

Figure 6.1 plots the marginal posterior $p(\rho, \sigma^2|y_{0:15})$, as approximated by SMC^2, run up to time 15. This figure illustrates the need for modelling nonparametrically the true likelihood as a function of θ, in order to estimate the variance of the estimated likelihood.

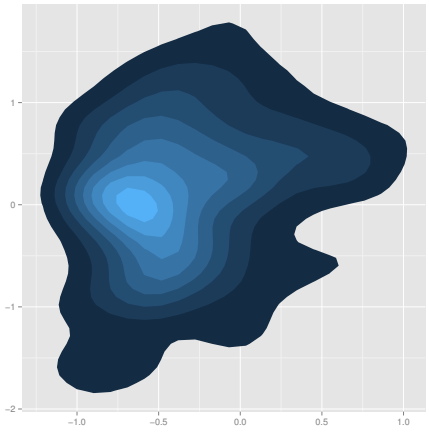


Figure 6.1: Marginal posterior $p(\sigma^2, \rho | y_{0:15})$, as approximated by SMC^2 run until $t = 15$, and linearly transformed so that axes are the two principal components.

For this model, sampling jointly from $\theta | x_{0:t}, y_{0:t}$ is difficult, but it is easy to perform a Gibbs step that leaves invariant $\theta | x_{0:t}, y_{0:t}$, as the full conditionals of each component (e.g. $\mu | \sigma, \rho, x_{0:t}, y_{0:t}$ and so on) are standard distributions. Let's call 'full PG' Algorithm 27, where Step 2 consists of this Gibbs step for $\theta | x_{0:t}, y_{0:t}$; and conversely let's call 'partial PG' Algorithm 27 with $\tilde{\theta} = \theta$ in Step 2 (θ is not updated).

We compare four versions of SMC^2: (a) the standard version, as proposed in Chopin et al. [2013a] (i.e. Step (c) of Algorithm 25 is a PMMH step, and that step is followed by an exchange step to double N_x when the acceptance rate of PMMH is below 20%); (b) the same algorithm, except that an exchange step is systematically performed after Step (c), and N_x is set to the value obtained with our non-parametric approach (see Section 6.3.2); (c) the version developed in this chapter, with full PG steps (and N_x updated through the non-parametric procedure); (d) the same algorithm, but with partial PG steps, followed by 3 PMMH steps to update θ.

The point of Algorithm (b) is to show that adapting N_x too often during the course of the algorithm is not desirable when using the exchange step, as this leads to too much variance. The point of Algorithm (d) is to see how our approach performs when sampling from $\theta | x_{0:t}, y_{0:t}$ (either independently or through MCMC) is not feasible.

Figure 6.2 plots the evolution of N_x over time for the four SMC^2 algorithms. One sees that, for these model and dataset, the CPU cost of the standard SMC^2 algorithm is quite volatile, as N_x increases very quickly in certain runs. In fact certain

runs are incomplete, as they were stopped when the CPU time exceeded 10 hours. On the other hand, the CPU cost of other versions is more stable across runs, and, more importantly, quite lower.

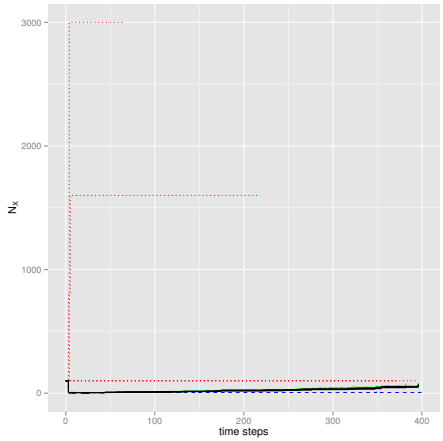


Figure 6.2: Evolution of N_x over time for 5 runs of the four considered SMC^2 algorithms; red dotted line is Algorithm (a), blue dashed is (b), black solid is (c), green double-dashed is (d). Results of (c) and (d) are nearly undistinguishable.

Figure 6.3 plots the empirical variance of the estimated marginal likelihood (evidence, $p(y_{0:t})$), normalised with the running time up to time step t. One observes that version (c) does quite better than (d), and far much better than (a). Results from Algorithm (b) were to variable to be included.

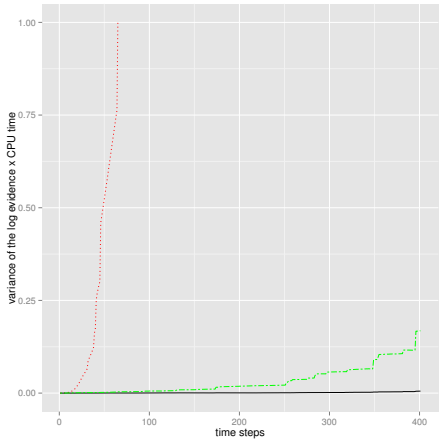


Figure 6.3: Empirical variance of estimated marginal likelihood $p(y_{0:t})$ multiplied by average CPU time; same legend as Figure 6.2, results from Algorithm (b) are omitted.

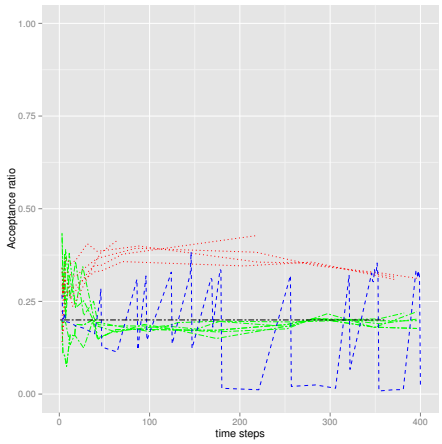


Figure 6.4: PMMH acceptance rate across time; same legend as Figure 6.2. Black line marks 20% target.

Figure 6.4 plots the acceptance rate of PMMH steps for Algorithms (a), (b) and (d). (Recall that Algorithm (c) does not perform PMMH steps). Note the poor performance of Algorithm (b). Figure 6.5 compares the box-plots of posterior estimates of σ at final time T, obtained from several runs of Algorithms (c) and (d). Algorithm (c) shows slightly less variability, while being 30% faster

on average. One sees that the improvement brought by ability to sample from $\theta|x_{0:t}, y_{0:t}$ is modest here for parameter estimation, but recall that in Figure 6.3, the improvement was more substantial.

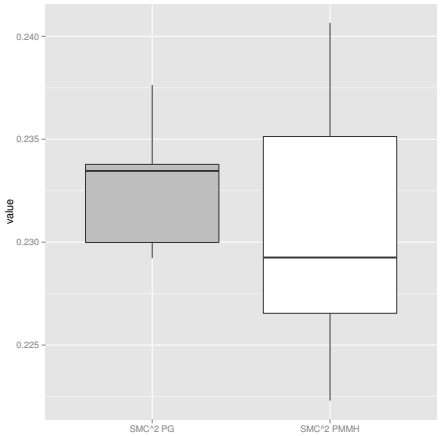


Figure 6.5: Box-plots of posterior estimate of parameter σ at final time T, over repeated runs of Algorithm (c) (left panel) and Algorithm (d) (right panel).

Bibliography

S. L. Adler. Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D*, 23(12):2901, 1981.

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *J. Am. Statist. Assoc.*, 88(422):669–79, 1993.

P. Alquier. Pac-bayesian bounds for randomized empirical risk minimizers. 17(4): 279–304, 2008.

P. Alquier. Bayesian methods for low-rank matrix estimation: short survey and theoretical study. In S. Jain, R. Munos, F. Stephan, and T. Zeugmann, editors, *Algorithmic Learning Theory*. Springer - Lecture Notes in Artificial Intelligence, 2014.

P. Alquier and G. Biau. Sparse single-index model. *J. Mach. Learn. Res.*, 14(1): 243–280, 2013.

P. Alquier and X. Li. Prediction of quantiles by statistical learning and application to GDP forecasting. In J.-G. Ganascia, P. Lenca, and J.-M. Petit, editors, *Discovery Science*. Springer - Lecture Notes in Artificial Intelligence, 2012.

C. Andrieu and G.O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009. doi: 10.1214/07-AOS574.

C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statist. Comput.*, 18 (4):343–373, 2008. doi: 10.1007/s11222-008-9110-y.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo. *J. R. Statist. Soc. B*, 72:269–342, 2010a.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010b. doi: 10.1111/j.1467-9868.2009.00736.x.

C. Andrieu, A. Lee, and M. Vihola. Uniform Ergodicity of the Iterated Conditional SMC and Geometric Ergodicity of Particle Gibbs samplers. *ArXiv e-prints*, December 2013.

J.-Y. Audibert and O. Catoni. Robust linear least squares regression. *Ann. Statist.*, 39(5):2766–2794, 10 2011. doi: 10.1214/11-AOS918. URL `http://dx.doi.org/10.1214/11-AOS918`.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On markov chain monte carlo methods for tall data. *arXiv preprint arXiv:1505.02827*, 2015.

J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD Cup and Workshop 07*, 2007.

Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534, nov 2013. doi: 10.3150/12-bej414. URL `http://dx.doi.org/10.3150/12-BEJ414`.

C. M. Bishop. *Pattern Recognition and Machine Learning*, chapter 10. Springer, 2006a.

C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006b.

P. Bissiri, C. Holmes, and S. Walker. A general framework for updating belief distributions. *arXiv preprint arXiv:1306.6430*, 2013.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities*. Oxford University Press, 2013.

F. Bretz, A. Genz, and L. A. Hothorn. On the numerical availability of multiple comparison procedures. *Biometrical journal*, 5:645–656, 2001.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensionnal Data*. Springer, 2011.

A. Bursh-Supan, V. Hajivassiliou, and L. J. Kotlikoff. Health, children and elderly arrangments: a multiperiod multinomial probit model with unobserved heterogeneity and autocorrelated errors. *Topics in the Economics of Aging*, pages 79–108, 1992.

R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2044061. URL `http://dx.doi.org/10.1109/TIT.2010.2044061`.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer Series in statistics, 2005.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

O. Catoni. *Statistical learning theory and stochastic optimization*, volume 1851 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2004. Lecture notes from the 31st Summer School on Probability Theory held in Saint-Flour, July 8–25, 2001.

O. Catoni. *PAC-Bayesian Supervised Classification*, volume 56. IMS Lecture Notes & Monograph Series, 2007.

O. Catoni. Challenging the empirical mean and empirical variance: A deviation study. *Ann. Inst. H. Poincaré Probab. Statist.*, 48(4):1148–1185, 11 2012. doi: 10.1214/11-AIHP454. URL `http://dx.doi.org/10.1214/11-AIHP454`.

V. Chernozhukov and H. Hong. An MCMC approach to classical estimation. *Journal of Econometrics*, 115(2):293–346, 2003.

Hugh Chipman, Edward I George, and Robert E McCulloch. *The practical implementation of Bayesian model selection*, pages 65–134. 2001.

N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89 (3):539–551, 2002a.

N. Chopin. A sequential particle filter for static models. *Biometrika*, 89:539–552, 2002b.

N. Chopin. Fast simulation of truncated Gaussian distributions. *Statist. Comput.*, 21(2):275–288, 2011a. ISSN 0960-3174. doi: 10.1007/s11222-009-9168-1.

N. Chopin. Fast simulation of truncated gaussian distributions. *Statist. Comput.*, 21(2):275–288, 2011b.

N. Chopin and C. Shaeffer. Sequential monte carlo on large binary sampling spaces. *Statist. Comput.*, 2011.

N. Chopin, P. Jacob, and O. Papaspiliopoulos. SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. *J. R. Statist. Soc. B*, 75(3):397–426, 2013a.

N. Chopin, O. Papaspiliopoulos, and P. E. Jacob. SMC^2: an efficient algorithm for sequential analysis of state space models. *J. R. Statist. Soc. B*, 75(3):397–426, 2013b.

J. A. Christen, C. Fox, D. A. Pérez-Ruiz, and M. Santana-Cibrian. On optimal direction Gibbs sampling. *arXiv preprint arXiv:1205.4062*, 2012.

S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and empirical minimization of U-statistics. *Ann. Stat.*, 36(2):844–874, 04 2008a.

S. Clémençon, V.C. Tran, and H. De Arazoza. A stochastic SIR model with contact-tracincing: large population limits and statistical inference. *Journal of Biological Dynamics*, 2(4):392–414, 2008b.

G. Consonni and J.M. Marin. Mean-field variational approximate Bayesian inference for latent variable models. *Comput. Stat. Data Anal.*, 52(2):790–798, 2007. doi: 10.1016/j.csda.2006.10.028. URL `http://dx.doi.org/10.1016/j.csda.2006.10.028`.

John D. Cook. Time exchange rate. *The Endeavour (blog)*, 2014. URL `http://www.johndcook.com/blog/2014/08/17/time-exchange-rate/`.

C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *NIPS*, volume 9, 2003.

A. S. Dalalyan and A. B. Tsybakov. Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity. *Machine Learning*, 72:39–61, 2008.

A. S. Dalalyan and A. B. Tsybakov. Sparse regression learning by aggregation and Langevin Monte-Carlo. *Journal of Computer and System Science*, 78(5):1423–1443, 2012.

P. Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996a.

P. Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996b.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006a.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Statist. Soc. B*, 68(3):411–436, 2006b. ISSN 1467-9868.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

R. Douc, O. Cappe, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proc. 4th Int. Symp. Image and Signal Processing and Analysis ISPA 2005*, pages 64–69, 2005.

A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

A. Doucet, M. Briers, and S. Senecal. Efiicient Block Sampling Strategies for Sequential Monte Carlo. *J. Comput. Graph. Statist.*, 15(3):693–711, 2006.

A. Doucet, M. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *ArXiv preprint*, October 2012.

S. Duane, D. Kennedy, A., B. J. Pendelton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, september 1987.

C. Dubarry and R. Douc. Particle approximation improvement of the joint smoothing distribution with on the fly variance estimation. *arXiv:1107.5524v1*, pages 1–19, June 2011.

Mohammad Emtiyaz Khan, Aleksandr Aravkin, Michael Friedlander, and Matthias Seeger. Fast dual variational inference for non-conjugate latent gaussian models. In *Proceedings of The 30th International Conference on Machine Learning*, pages 951–959, 2013.

C. Faes, J. Ormeros, and M. Wand. Variational Bayesian Inference for Parametric and Nonparametric Regression With Missing Data. *J. Am. Statist. Assoc.*, 106 (495):959–971, September 2011.

P. Fearnhead, O. Papaspiliopoulos, G.O. Roberts, and A. Stuart. Random weight particle filtering of continuous time processes. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72:497–513, 2010.

D. Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1): 27–38, 1993.

C. Flecher, D. Allard, and P. Naveau. Truncated skew-normal distributions: Estimation by weighted moments and application to climatic data. Technical Report 39, Institut National de la Recherche Agronomique, 2009.

Y. Freund, R. Iyer, R.E Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

Sylvia Frühwirth-Schnatter and Rudolf Frühwirth. Data augmentation and mcmc for binary and multinomial logit models. In *Statistical Modelling and Regression Structures*, pages 111–132. Physica-Verlag HD, Dec 2009. doi: 10.1007/978-3-7908-2413-1_7.

Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2006.

Andrew Gelman, Aleks Jakulin, Maria Grazia Pittau, and Yu-Sung Su. A weakly informative default prior distribution for logistic and other regression models. *Ann. Appl. Stats.*, 2(4):1360–1383, 2008. ISSN 1932-6157. doi: 10.1214/08-AOAS191. URL `http://dx.doi.org/10.1214/08-AOAS191`.

A. Genz. Numerical Computation of Multivariate Normal Probabilities. *J. Comput. Graph. Statist.*, 1(2):141–149, June 1992.

A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities*, volume 195 of *Lecture Notes in Statistics*. Springer, 2009.

Edward I. George and Robert E. McCulloch. Variable selection via gibbs sampling. *J. Am. Statist. Assoc.*, 88(423):881–889, sep 1993a. doi: 10.1080/01621459.1993.10476353. URL `http://dx.doi.org/10.1080/01621459.1993.10476353`.

E.I. George and R.E. McCulloch. Variable selection via Gibbs sampling. *J. Am. Statist. Assoc.*, 88(423):pp. 881–889, 1993b.

J. Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints. *Computing Science and Statistics*, 23: 571–578, 1991.

G. J. Gibson, C. A. Glasbey, and D. A. Elston. Monte-carlo evalution of multivariate normal integrals and sensitivity to variate ordering. *Advances in Numerical Methods & applications*, pages 120–126, 1994.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Statist. Soc. B*, 73(2):123–214, 2011.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

Robert B. Gramacy and Nicholas G. Polson. Simulation-based regularized logistic regression. 7(3):567–590, Sep 2012. doi: 10.1214/12-ba719.

P. J. Green. Reversible Jump Markov Chain Monte Carlo computation and Bayesian Model Determination. *Biometrika*, 82(4):711–732, 1995.

P. J. Green, K. Latuszynski, M. Pereyra, and C. P. Robert. Bayesian computation: a perspective on the current state, and sampling backwards and forwards. Preprint arXiv:1502.01148, 2015.

B. Guedj and P. Alquier. PAC-Bayesian estimation and prevision in sparse additive models. *Electronic Journal of Statistics*, 7:264–291, 2013.

V. Hajivassiliou, D. McFadden, and P. Ruud. Simulation of multivariate normal rectangle probabilities and their derivatives theoretical and computational results. *Journal of Econometrics*, 72(1-2):85–134, May–June 1996.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

D. Hernandez-Lobato, J. Hernandez-Lobato, and P. Dupont. Generalized Spike-and-Slab Priors for Bayesian Group Feature Selection Using Expectation Propagation . *J. Mach. Learn. Res.*, 14:1891–1945, 2013.

Y. Hochberg and A. C. Tamhane. *Multiple comparison procedures*. John Wiley & Sons, Inc., 1987.

W. Hoeffding. Probability Inequalities for Sums of Random Variables. *Ann. Math. Stat.*, 10:293–325, 1948.

M. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian monte carlo. *J. Mach. Learn. Res.*, page (in press), 2013.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. 1(1):145–168, 2006.

Wolfgang Hörmann and Josef Leydold. Quasi importance sampling. Technical report, 2005.

P. Jacob, C. P. Robert, and M. H. Smith. Using parallel computation to improve independent metropolis–hastings based estimation. *J. Comput. Graph. Statist.*, 20(3):616–635, Jan 2011. doi: 10.1198/jcgs.2011.10167.

P.E. Jacob, L. Murray, and S. Rubenthaler. Path storage in the particle filter. *Statist. Comput.*, pages 1–10, 2013. ISSN 0960-3174. doi: 10.1007/s11222-013-9445-x. URL `http://dx.doi.org/10.1007/s11222-013-9445-x`.

A. Jasra, D. Stephens, and C. Holmes. On population-based simulation for static inference. *Statist. Comput.*, 17(3):263–279, 2007.

A. Jasra, D. Stephens, A. A. Doucet, and T. Tsagaris. Inference for Lévy driven stochastic volatility models via Sequential Monte Carlo. *Scand. J. of Statist.*, 38(1), 2011a.

Ajay Jasra, David A Stephens, Arnaud Doucet, and Theodoros Tsagaris. Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38(1):1–22, 2011b.

W. Jiang and M. A. Tanner. Gibbs posterior for variable selection in high-dimensional classification and data mining. *The Annals of Statistics*, 36(5): 2207–2231, 2008.

M. I. Jordan, Z. Ghahrapani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, (37):183–233, 1999.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *The Journal of Machine Learning Research*, 12:3227–3257, 2011.

Ata Kabán. On bayesian classification with laplace priors. *Pattern Recognition Letters*, 28(10):1271–1282, 2007. doi: 10.1016/j.patrec.2007.02.010.

M. Keane. Simulation estimation for panel data models with limited dependent variables. MPRA Paper 53029, University Library of Munich, Germany, 1993.

M. E. Khan. Decoupled variational Gaussian inference. In *Advances in Neural Information Processing Systems*, pages 1547–1555, 2014.

V. Koltchinskii, K. Lounici, and A. B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

Demetris Lamnisos, Jim E. Griffin, and Mark F. J. Steel. Adaptive Monte Carlo for Bayesian variable selection in regression models. *J. Comput. Graph. Statist.*, 22(3):729–748, 2013. ISSN 1061-8600. doi: 10.1080/10618600.2012.694756. URL `http://dx.doi.org/10.1080/10618600.2012.694756`.

N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 601–608. ACM, 2009.

G. Lecué. Méthodes d'agrégation: optimalité et vitesses rapides. Ph.D. thesis, Université Paris 6, 2007.

Anthony Lee, Christopher Yau, Michael B. Giles, Arnaud Doucet, and Christopher C. Holmes. On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *J. Comput. Graph. Statist.*, 19 (4):769–789, jan 2010. doi: 10.1198/jcgs.2010.10039. URL `http://dx.doi.org/10.1198/jcgs.2010.10039`.

Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling (Springer Series in Statistics)*. Springer, February 2009. ISBN 0387781641.

J. P. LeSage, Pace R. K., N. Lam, R. Campanella, and Liu X. New Orleans buisness recovery in the aftermath of huricane Katrina. *App. Stat.*, 174(4): 1007–1027, October 2011.

Y. J. Lim and Y. W. Teh. Variational Bayesian approach to movie rating prediction. *Proceedings of KDD Cup and Workshop*, 7:15–21, 2007.

D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2002.

T. T. Mai and P. Alquier. A Bayesian approach for matrix completion: optimal rate under general sampling distribution. *Electronic Journal of Statistics*, 9: 823–841, 2015.

E. Mammen and A. Tsybakov. Smooth discrimination analysis. *Ann. Stat.*, 27(6): 1808–1829, 12 1999.

P. Massart. *Concentration Inequalities and Model Selection*, volume 1896. Springer Lecture Notes in Mathematics, 2007.

D. A. McAllester. PAC-Bayesian model averaging. In *Proceedings of of the Twelth Annual Conference On Computational Learning Theory, Santa Cruz, California (Electronic)*, pages 164–170. ACM, New-York, 1999.

D.A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM, 1998.

S. Meyn and R. L. Tweedie. *Markov chains and Stochastic Stability*. Cambridge University Press, 2nd edition, 2009.

T. Minka. Expectation Propagation for approximate Bayesian inference. In *Proc. 17th Conf. Uncertainty Artificial Intelligence*, UAI '01, pages 362–369. Morgan Kaufmann Publishers Inc., 2001a.

T.P. Minka. Expectation Propagation for approximate Bayesian inference. *Proceedings of Uncertainty in Artificial Intelligence*, 17:362–369, 2001b.

T. Minwa, A. J. Hayter, and S. Kuriki. The evaluation of general non-centered orthant pro. *J. R. Statist. Soc. B*, 65:223–234, 2003.

T. J Mitchell and J. Beauchamp. Bayesian variable selection in linear regression. *J. Am. Statist. Assoc.*, 83(404):1023–1032, 1988.

M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, page 51, 2010a.

R. Neal. Annealed importance sampling. *Statist. Comput.*, 11(2):125–139, 2001a.

R. M. Neal. Annealed importance sampling. *Statist. Comput.*, 11:125–139, 2001b.

R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman & Hall / CRC Press, 2010b. URL `http://www.cs.utoronto.ca/~radford/ftp/ham-mcmc.pdf`.

Y. Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

H. Nickisch and C.E. Rasmussen. Approximations for Binary Gaussian Process Classification. *J. Mach. Learn. Res.*, 9(10):2035–2078, October 2008.

H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bulletin of the american mathematical society*, 84(6):957–1041, November 1978.

M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

M. Opper and O. Winther. Gaussian Processes for Classification: Mean-field Algorithms. *Neural Computation*, 12(11):2655–2684, November 2000.

A. Pakman and L. Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *arXiv:1208.4118*, pages 1–30, 2012.

M.D. Pandey. An effective approximation to evaluate multinormal integrals. *Structural Safety*, 20(1):51–67, 1998.

G. Parisi. *Statistical field theory.* Addison-Wesley, New-York, 1988.

M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, 94(446):590–599, June 1999.

Nicholas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, Dec 2013. doi: 10.1080/01621459.2013.829001. URL `http://dx.doi.org/10.1080/01621459.2013.829001`.

A. Prekopa. On probabilistic constrained programming. In *Proceedings of the Princeton symposium on mathematical programming*, pages 113–138. Princeton, New Jersey: Princeton University Press, 1970.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 2007.

C. Rasmussen and C. Williams. *Gaussian processes for Machine Learning.* MIT press, 2006.

James Ridgway. Computation of Gaussian orthant probabilities in high dimension. *(Minor revision) Statist. Comput.*., 2015.

S. Robbiano. Upper bounds and aggregation in bipartite ranking. *Elec. J. of Stat.*, 7:1249–1271, 2013.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

C. P. Robert. Simulation of truncated normal variables. *Statist. Comput.*, 5(2): 121–125, 1995.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods, 2nd ed.* Springer-Verlag, New York, 2004a.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*, chapter 9. Springer, 2004b.

Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation.* Springer Science & Business Media, 2007.

G. Roberts and J. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *J. R. Statist. Soc. B*, 60(1):255–268, 1998.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statist. Science*, 16(4):351–367, 2001. ISSN 08834237. doi: 10.1214/ss/1015346320.

Gareth O Roberts and Jeffrey S Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

V. Ročková and E. George. Emvs: The EM approach to bayesian variable selection. *J. Am. Statist. Assoc.*, 2013.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J. R. Statist. Soc. B*, 71(2):319–392, 2009.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

C. Schäfer and N. Chopin. Sequential monte carlo on large binary sampling spaces. *Statistics and Computing*, pages 1–22, 2011.

Christian Schäfer. *Monte Carlo methods for sampling high-dimensional binary vectors*. PhD thesis, Université Paris Dauphine, 2012.

Steven L. Scott, Alexander W. Blocker, and Fernando V. Bonassi. Bayes and big data: The consensus monte carlo algorithm. In *Bayes 250*, 2013.

M. Seeger. Expectation propagation for exponential families. Technical report, U. of California, 2005a.

M. Seeger. Expectation Propagation for Exponential Families. Technical report, Univ. California Berkeley, 2005b.

Babak Shahbaba, Shiwei Lan, Wesley O Johnson, and Radford M Neal. Split hamiltonian monte carlo. *Statist. Comput.*, pages 1–11, 2011. doi: 10.1007/s11222-012-9373-1. URL `http://arxiv.org/pdf/1106.5941.pdf`.

J. Shawe-Taylor and R.C. Williamson. A PAC analysis of a Bayesian estimator. In *Proc. conf. Computat. learn. theory*, pages 2–9. ACM, 1997.

J. Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–860, 2006.

Marc A. Suchard, Quanli Wang, Cliburn Chan, Jacob Frelinger, Andrew Cron, and Mike West. Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *J. Comput. Graph. Statist.*, 19 (2):419–438, jan 2010. doi: 10.1198/jcgs.2010.10016. URL `http://dx.doi.org/10.1198/jcgs.2010.10016`.

T. Suzuki. Convergence rate of Bayesian tensor estimator: Optimal rate without restricted strong convexity. arXiv preprint arXiv:1408.3092 (accepted by ICML2015), 2014.

L. Tierney, R. E. Kass, and J. B. Kadane. Fully exponential Laplace approximations to expectations and variances of non-positive functions. *J. Am. Statist. Assoc.*, 84:710–716, 1989.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *J. Am. Statist. Assoc.*, 81(393):82–86, 1986.

K. E. Train. *Discrete Choice methods with simulation*. Cambridge University Press, 2009.

A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

A. Van der Vaart. *Asymptotic statistics*. Cambridge university press, 1998.

A.W. van der Vaart and J.H. van Zanten. Adaptive Bayesian estimation using a Gaussian random field with inverse Gamma bandwidth. *Ann. Stat.*, pages 2655–2675, 2009.

M. A.J. Van Gerven, B. Cseke, F. P. de Lange, and T. Heskes. Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage*, 50:150–161, 2010.

Xiangyu Wang and David B Dunson. Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

O. Wintenberger. Deviation inequalities for sums of weakly dependent time series. *Electronic Communications in Probability*, 15:489–503, 2010.

L. Yan, R. Dodier, M. Mozer, and R. Wolniewicz. Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. *Proc. 20th Int. Conf. Mach. Learn.*, pages 848–855, 2003.

Y. Yang. Aggregating regression procedures to improve performance. *Bernoulli*, 10:25–47, 2004.

G. Yao and U Bockenholt. Bayesian estimation of Thurstonian ranking models based on the Gibbs sampler. *British Journal of Mathematical and Statistical Psychology*, 52:79–92, 1999.

A. Yuille. Belief Propagation, Mean-field and the Bethe approximation. Technical report, Dept. Statistics UCLA.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.

T. Zhang. Information theoretical upper and lower bounds for statistical estimation. *IEEE Transaction on Information Theory*, 52:1307–1321, 2006.

M. Zhou, C. Wang, M. Chen, J. Paisley, D. Dunson, and L. Carin. Nonparametric bayesian matrix completion. *Proc. IEEE SAM*, 2010.