Inference and applications for topic models

Christophe Dupuy

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'Ecole Normale Supérieure

Inférence et Applications pour les Modèles Thématiques

Inference and Applications for Topic Models

Ecole doctorale n°386

Sciences Mathématiques de Paris Centre

Spécialité INFORMATIQUE

COMPOSITION DU JURY :

M. CAPPE Olivier
CNRS – LIMSI, Président du jury

M. CARON François
University of Oxford, Rapporteur

M. TITSIAS Michalis
Athens University of Economics and Business, Rapporteur

M. D'ASPREMONT Alexandre
CNRS – ENS – INRIA, Membre du jury

M. DIOT Christophe
Safran, Membre du jury

M. PEREZ Patrick
Technicolor, Membre du jury

Soutenue par Christophe Dupuy
Le 30 Juin 2017

Dirigée par **Francis Bach**

ENS ÉCOLE NORMALE SUPÉRIEURE | PSL RESEARCH UNIVERSITY PARIS

technicolor

PhD thesis - École Normale Supérieure de Paris
February 2014 — July 2017

Inference and Applications for Topic Models

Christophe Dupuy

Advisors: Francis Bach & Patrick Pérez

ABSTRACT

Most of current recommendation systems are based on ratings (i.e. numbers between 0 and 5) and try to suggest a content (movie, restaurant...) to a user. These systems usually allow users to provide a text review for this content in addition to ratings. It is hard to extract useful information from raw text while a rating does not contain much information on the content and the user. In this thesis, we tackle the problem of suggesting personalized readable text to users to help them make a quick decision about a content.

More specifically, we first build a topic model that predicts personalized movie description from text reviews. Our model extracts distinct qualitative (i.e., which convey opinion) and descriptive topics by combining text reviews and movie ratings in a joint probabilistic model. We evaluate our model on an IMDB dataset and illustrate its performance through comparison of topics.

We then study parameter inference in large-scale latent variable models, that include most topic models. We propose a unified treatment of online inference for latent variable models from a non-canonical exponential family, and draw explicit links between several previously proposed frequentist or Bayesian methods. We also propose a novel inference method for the frequentist estimation of parameters, that adapts MCMC methods to online inference of latent variable models with the proper use of local Gibbs sampling. For the specific latent Dirichlet allocation topic model, we provide an extensive set of experiments and comparisons with existing work, where our new approach outperforms all previously proposed methods.

Finally, we propose a new class of determinantal point processes (DPPs) which can be manipulated for inference and parameter learning in potentially sublinear time in the number of items. This class, based on a specific low-rank factorization of the marginal kernel, is particularly suited to a subclass of continuous DPPs and DPPs defined on exponentially many items. We apply this new class to modelling text documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions, which is made possible with no approximation for our class of DPPs. We present an application to document summarization with a DPP on 2^{500} items, where the summaries are composed of readable sentences.

KEYWORDS: topic models, online learning, latent variable models, unsupervised learning, determinantal point processes, latent Dirichlet allocation.

RÉSUMÉ

La plupart des systèmes de recommandation actuels se base sur des évaluations sous forme de notes (i.e., chiffre entre 0 et 5) pour conseiller un contenu (film, restaurant...) à un utilisateur. Ce dernier a souvent la possibilité de commenter ce contenu sous forme de texte en plus de l'évaluer. Il est difficile d'extraire de l'information d'un texte brut tandis qu'une simple note contient peu d'information sur le contenu et l'utilisateur. Dans cette thèse, nous tentons de suggérer à l'utilisateur un texte lisible personnalisé pour l'aider à se faire rapidement une opinion à propos d'un contenu.

Plus spécifiquement, nous construisons d'abord un modèle thématique prédisant une description de film personnalisée à partir de commentaires textuels. Notre modèle sépare les thèmes qualitatifs (i.e., véhiculant une opinion) des thèmes descriptifs en combinant des commentaires textuels et des notes sous forme de nombres dans un modèle probabiliste joint. Nous évaluons notre modèle sur une base de données IMDB et illustrons ses performances à travers la comparaison de thèmes.

Nous étudions ensuite l'inférence de paramètres dans des modèles à variables latentes à grande échelle, incluant la plupart des modèles thématiques. Nous proposons un traitement unifié de l'inférence en ligne pour les modèles à variables latentes à partir de familles exponentielles non-canoniques et faisons explicitement apparaître les liens existants entre plusieurs méthodes fréquentistes et Bayesiennes proposées auparavant. Nous proposons aussi une nouvelle méthode d'inférence pour l'estimation fréquentiste des paramètres qui adapte les méthodes MCMC à l'inférence en ligne des modèles à variables latentes en utilisant un échantillonnage de Gibbs local. Pour le modèle thématique d'allocation de Dirichlet latente, nous fournissons une vaste série d'expériences et de comparaisons avec des travaux existants dans laquelle notre nouvelle approche est plus performante que les méthodes proposées auparavant.

Enfin, nous proposons une nouvelle classe de processus ponctuels déterminantaux (PPD) qui peut être manipulée pour l'inférence et l'apprentissage de paramètres en un temps potentiellement sous-linéaire en le nombre d'objets. Cette classe, basée sur une factorisation spécifique de faible rang du noyau marginal, est particulièrement adaptée à une sous-classe de PPD continus et de PPD définis sur un nombre exponentiel d'objets. Nous appliquons cette classe à la modélisation de documents textuels comme échantillons d'un PPD sur les phrases et proposons une formulation du maximum de vraisemblance conditionnel pour modéliser les proportions de thèmes, ce qui est rendu possible sans aucune approximation avec notre classe de PPD. Nous présentons une application à la synthèse de documents avec un PPD sur 2^{500} objets, où les résumés sont composés de phrases lisibles.

MOTS CLÉS: modèles thèmatiques, apprentissage en ligne, modèles à variables latentes, apprentissage non supervisé, processus ponctuels determinantaux, allocation de Dirichlet latente.

ACKNOWLEDGEMENT

I am lucky enough to say that many persons have contributed directly or indirectly to this thesis and I would like to thank all of them. I both thank all the persons I have met in a professional context during my PhD and my friends. As I know most people will only read this chapter, I try to be as exhaustive as possible.

I would like to first thank my advisor, Francis Bach. Not only has he been my advisor, but he has also taught me the long process to achieve high quality research. He has always been available, has always taken the time to set a meeting when I needed despite his very busy agenda and to answer my mails despite the dozens (probably more) he receives each day. Especially during my last year, he spent a lot of time guiding me during my job search. He found the right words to reconfort me after paper rejections. He conveyed his appetite for machine learning and research to me from the MVA class until now. I am infinitely grateful to him.

I perfectly remember my first meeting with Christophe Diot, before my master's internship. I would not be writing these words without him. He conveyed his passion for research and his determination to me. Our meetings were hardly restful but always very enriching and I thank Christophe for these challenging exchanges. I also acknowledge support from Technicolor allowing me to make a CIFRE PhD and staying in contact with industry during these three years.

If I were to start all over again, it is not certain I would have met Patrick Pérez. Yet, I had this incredible chance to work and exchange with him during my PhD. I thank Patrick for all the time he devoted to this work between Paris and Rennes.

I thank the reviewers of this manuscript François Caron and Michalis Titsias for their time and their enlightening reports. I also thank the jury members Olivier Cappé and Alexandre D'Aspremont.

I would also like to thank the expermimented and smart INRIA researchers I had the chance to meet during my stay: Sylvain Arlot, Simon Lacoste-Julien, Guillaume Obozinski, Alexandre D'Aspremont, Pierre Gaillard. During the seminars and the always friendly exchanges with them, they always fully dedicate their sharp eye for machine learning to Research and it has been very pleasant to rub shoulders with them.

I acknowledge support from the CIFAR program in Learning in Machines & Brains.

I would like to thank the Dupuy family to have supported me during these years. They've always believed in me even when I have my doubts and when I hold back from talking about it. I also thank the Labonne family for their

support, for all the invitations and for their always warm welcome at any sunny/rainy Sunday lunch.

From this line begins a long list of colleagues or friends (or both) I have worked or hanged out with (or both) during my PhD.

I would like to first thank all the office mates who bore my company during these years: Anton, Nastya*, Gül, Pascal, Alexandre, Mathieu. I would also to thank all the PhDs and Postdocs I got to know during my PhD: Nicolas F.*, Vincent, JB, Matthew, Qassem*, Julia, Guilhem*, Damien, Igor*, Antoine*, Aymeric, Nicolas B., Relja, Bala, Federico, Fabian

I would also like to thank my friends who took the time to understand and discuss the subject of my PhD, most often around a table full with appetizing food and drinks. I particularly thank Thibaut, Isa, Lucas, Prosper, Océane, Xavier, Marion, Fred, Anne, Louis.

As I always do, I saved the best for last: I would like to thank Claire to team up with me for all these years.

*Special thanks for the regular diverting coffee/lunch breaks

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

The amount of text and the number of documents available on the Internet have recently skyrocketed. The types and uses of documents are very diverse. For instance, reviews of products you may find in recommender systems (e.g., Yelp[1], Amazon[2], IMDB[3]) convey an opinion and are crowd-sourced. On the other hand, newspapers or scientific papers are neutral and written by professionals. We identify other sources of text. Websites like Quora[4] or Stackoverflow[5] are used as questions and answers (Q & A) platforms where users expose questions or issues they are trying to solve (e.g., math problems) while other users propose possible solutions. Search engines represent another source of text. The user makes a request—usually formulated with text—and the aim of a search engine is to suggest the most relevant content(s) to this request. The recent emergence of personal assitants (e.g., Siri[6], "Ok Google"[7], Alexa[8]) gives us a glimpse of new sources of text. These devices turn vocal commands into text requests as inputs for search engines.

A common problem for these applications is to process a huge amount of text (potentially millions or billions of documents or requests). In particular, the extraction of useful information for a given task is proving to be difficult. Note that the definition of usefulness clearly depends on the application. For recommender systems, the user intends to choose a content from a list as quickly as possible. Useful reviews could be those which help users to make a quick opinion about a content. For search engines, computing a distance between any text request and any available content would be useful. In practice, this distance is learned from examples of requests and enables the engine to suggest efficiently the closest content to the user's query.

In this context, Technicolor[9] is a worldwide leader in the media and entertainment sector. The company is specialized in movie creation, production and distribution. It delivers solutions for content management (such as creation, imaging, finishing, preparation) and offers a wide range of services for providing digital entertainment (such as movies) at home through Pay-TV operators and network service providers. Consequently, the motivations for this work

[1] http://www.yelp.com
[2] http://www.amazon.com
[3] http://www.imdb.com
[4] http://www.quora.com
[5] http://stackoverflow.com/
[6] http://www.apple.com/ios/siri/
[7] http://madeby.google.com/home/
[8] http://www.amazon.com/echo
[9] http://www.technicolor.com/

were to look for new methods including text to recommend movies for users at home.

In this thesis we focus on services including crowd-sourced reviews such as Yelp, IMDB, the Fork[10], Amazon etc. For this type of service, the main issue is to help users to assess contents (restaurants, movies, products) as fast as possible. While many recommender systems are based on rating prediction [Ricci et al., 2011], the solution of most commercial services to this issue is to reduce the quantity of information (reviews, ratings) absorbed by the user before making a decision. We can identify various strategies of commercial services to implement this solution. While all the services propose the "Sort by" feature for reviews, they also have an additional specific feature on top of that when browsing a particular content:

- On Yelp, selected sentences are displayed based on key-words they contain. The key-words are highlighted and correspond to specific aspects (served dishes or ingredients, location, quality of service, etc.) of the restaurant. The platform also allows the user to characterize any existing review as *useful*, *funny* or *cool*. An user can then quickly assess the "quality" of a review before reading it, based on the number of *useful*, *funny* or *cool* already assigned to this review.

- On IMDB, the *useful* feature is also implemented, but through a yes or no question: *Was the above review useful to you?*. The reviews can be ranked in the "Sort by" feature by computing the proportion of useful assignments for each review (i.e., the score of a review is the number of "yes" answers divided by the total number of answers for this review; the reviews with maximal scores will be first displayed).

- On Amazon, the *Top customer reviews* (reviews with the highest rating), *Top critical reviews* (reviews with the lowest rating) and the *Most recent customer reviews* are all displayed on the first page of the product and it is easy to access to the reviews with a given rating. The *useful* feature is also implemented—as in Yelp—to quickly assess the quality of a review.

- On the Fork, the platform allows the user to assess a given restaurant with a global rating and several "sub-ratings" to evaluate the different aspects of the restaurants (*food*, *service*, *setting*, *value for price* among others). The user can also specify the *occasion* for visiting the restaurant (*on your own*, *with friends*, *with family*, *romantic* or *business*). As a result, the average rating of each aspect is displayed on the homepage of a given restaurant so the user has a quick overview of the aspects for this restaurant. For more details on restaurant, the user can also filter the reviews by *occasion*.

We remark that in any case, the selected reviews or features displayed are not personalized (i.e., do not depend on the user browsing the website) and the user

[10] http://www.thefork.com

usually needs to manually select and parse the particular reviews she needs to read to make her final opinion. Even if the selection process is greatly simplified by the different platforms, the effort made by the user may still be significant.

Given these commercial solutions, our goal is to automate and personalized the suggestions to the users, based on the history of reviews and ratings of many different users. More precisely, the ultimate goal of this work is to build a model that would automatically suggest a personalized list of contents to the user together with a (short) personalized readable text attached to each content describing the (as accurate as possible) opinion of the user on this content. In this thesis, we build a line of work towards this ultimate goal, taking advantage of substantial existing work on information retrieval.

1.1 INTRODUCTION TO INFORMATION RETRIEVAL

The term "information retrieval" (IR) was first used by Mooers [1950]. It gathers all the techniques to extract information from a large corpus of text documents. The retrieved information can take the form of meta-data, latent representation (as in topic models), documents or other types of contents (as in search engines). The applications derived from IR are numerous [Aggarwal and Zhai, 2012]. For instance, IR covers text classification [Nigam et al., 1999, Rennie, 2001], document clustering [Pereira et al., 1993] or semantic features extraction [Deerwester et al., 1990, Dumais, 1994, Hofmann, 1999b,a]. Given our initial objective of text recommendation to users, we focus on semantic features extraction—also known as topic models. This particular field of IR covers techniques able to identify and gather words that have similar meanings from a corpus of documents. In this section, we present the document representation mostly used in topic models and the first techniques to extract semantic features from text.

1.1.1 Bag-of-words representation of document

The bag-of-words representation is the most widely used representation of documents and was first introduced by Harris [1954]. Given a corpus of D documents $\mathcal{C} = \{X^1, \dots, X^D\}$, we denote V the size of the vocabulary, i.e., the number of different words used in the corpus. In the bag-of-words representation, the document i of $\mathcal{C}$ is represented as a vector $X^i \in \mathbb{R}^V$, with:

$$\forall v \in [\![1, V]\!], (X^i)_v = \begin{cases} n_v^i > 0 & \text{if word } v \text{ occurs in the } i\text{-th document of } \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

In this representation, n_v^i represents the "score" of word v in document i. A first natural approach consists in setting n_v^i to the frequency of word v in document i, but other more evolved scores are used in practice, such as term frequency-inverse document frequency (TF-IDF) [Hiemstra, 2000]. In this representation, neither the order of words nor the structure of the documents (e.g., sentences,

punctuation, etc.) are taken into consideration. The co-occurrences of words are the only structure represented with bag-of-words, which is often enough information to extract semantics, i.e., to gather words with similar (semantic) meaning.

It is possible to model documents with more complex representations but a lot of technical obstacles arise from these representations. A 2-gram is defined as a pair of consecutive words in a larger sequence of words (i.e., a sentence or a document). In a 2-gram representation, we consider all the single words and pairs of consecutive words (i.e., the sequences of 1 or 2 consecutive words) that occur in corpus $\mathcal{C}$. If we denote M the number of such sequences in corpus $\mathcal{C}$, the associated representation of document i is a vector $\hat{X}^i \in \mathbb{R}^M$ with $(\hat{X}^i)_m = n^i_m > 0$ if the sequence m occurs in document i and 0 otherwise, with a similar definition of n^i_m than n^i_v in bag-of-words. This representation is more complete because it models all the possible pairs of words occurring in corpus $\mathcal{C}$ in addition to single words, but it is very complicated to manipulate in practice as $M \gg V$. Another con of this representation is that even if M is large, only a small number of sequences actually conveys semantics. For instance, if we consider of corpus of long documents (e.g., newspapers with hundreds of words per documents), among the different pairs of consecutive words (i.e., 2-grams) in the corpus, only a very small portion appears more than once and among the pairs appearing more than once in the corpus, only a very few of them actually convey semantics. As a result, in a 2-gram representation, the signal to noise ratio is very low we have to make a lot more effort to extract semantics. It is even more difficult if we consider higher order n-grams (i.e., sequences of n consecutive words), with $n > 2$. See for instance Jelinek and Mercer [1980], Katz [1987], Kneser and Ney [1995].

1.1.2 First topic models

From the bag-of-words representations descrived above, the first IR models are algebraic models, in which documents are represented as vectors. One of the first applications is the vector space model [Salton et al., 1975], where the bag-of-words representation with TF-IDF score is used for document retrieval. The objective is similar to search engines, namely, return the (stored) document that best matches the (incoming) user's text query. The similarity between the user's query and each stored document is computed with the cosine similarity, i.e., as the (normalized) scalar product between the bag-of-words representations of the query and the document. This first approach is clearly not scalable: given a query, one needs to compute D scalar products of size V, where D is the number of stored documents and V is the size of the vocabulary. If the number of documents D is several billions and the vocabulary contains several thousands words, the time to return a document from a single query may be too long with this representation.

A first extension of this model is the latent semantic analysis (LSA) or indexing (LSI) model [Deerwester et al., 1990]. Given a corpus $\mathcal{C} = \{X^1, \dots, X^D\}$ where $X^i \in \mathbb{R}^V$ is the bag-of-words representation of document i, we consider the term-document matrix $T_D \in \mathbb{R}^{V \times D}$, defined as follows:

$$T_D = \left(\begin{bmatrix} \\ X^1 \\ \\ \end{bmatrix} \begin{bmatrix} \\ \cdots \\ \\ \end{bmatrix} \begin{bmatrix} \\ X^D \\ \\ \end{bmatrix} \right).$$

We reduce the dimension of the document representation by first computing the SVD decomposition of T_D: $T_D = P\text{diag}(\sigma) Q^\top$, where $P \in \mathbb{R}^{V \times r}$ and $Q \in \mathbb{R}^{D \times r}$ are the singular vectors of T_D and $\sigma \in \mathbb{R}^r$ is the vector of the non-zero singular values of T_D. In particular, we can represent document i of the corpus as a vector of size r using the following identity:

$$X^i = Px^i = P\text{diag}(\sigma)q^i,$$

where $q^i \in \mathbb{R}^r$ is the i-th row of Q and $x^i = \text{diag}(\sigma)q^i \in \mathbb{R}^r$ is the r-dimensional representation of document i. The vector $x^i \in \mathbb{R}^r$ corresponds to the weights given by document i to the r left-singular vectors, i.e., the r columns of P. In other words, x^i corresponds to the coordinates of X^i in the vectorial space of dimension r in $\mathbb{R}^V$ with orthonormal basis the r columns of P. As the matrix P is orthogonal, we simply compute the r-dimensional representation x^i of document i from X^i as follows:

$$x^i = P^\top X^i.$$

This transformation corresponds to a change of basis and can thus be extended to any bag-of-words representation. Given a bag-of-words $X \in \mathbb{R}^V$, the corresponding r-dimensional transformation x is given by:

$$x = P^\top X.$$

In particular, if we consider the transformation of the bag-of-words representation of an incoming query, the cosine similarity between this query and any (stored) document of corpus $\mathcal{C}$ is a scalar product of size r (instead of V above).

However, as the rank r of the term-document matrix may be large (i.e., close to V), we can reduce even more the dimension of the transfomation. We first choose the K largest singular values $\sigma_K \in \mathbb{R}^K$ of T_D and the corresponding K singular vectors $P_K \in \mathbb{R}^{V \times K}$ and $Q_K \in \mathbb{R}^{D \times K}$. The matrix $\hat{T}_K = P_K\text{diag}(\sigma_K)(Q_K)^\top$ is the best (in Frobenius norm) rank-K approximation of T_D [Eckart and Young, 1936]. We then use the following approximation to compute the K-dimensional representation of document i:

$$X^i = P\text{diag}(\sigma)q^i \approx P_K\text{diag}(\sigma_K)(q_K)^i,$$

where $(q_K)^i \in \mathbb{R}^K$ is the i-th row of Q_K. Given this approximation and given that P_K is orthogonal, we obtain the transformation $\hat{x}^i \in \mathbb{R}^K$ of X^i with the following formulation:

$$\hat{x}^i = (P_K)^\top X^i.$$

The intuition is that the complete representation of X^i in the basis P is $x^i \in \mathbb{R}^r$. With this new representation $\hat{x}^i \in \mathbb{R}^K$, we only consider the K coordinates with the highest importance (which is conveyed by σ) to represent X^i. We use the same reasoning to extend this new transformation to any bag-of-words representation: $x = (P_K)^\top X \in \mathbb{R}^K$. The cosine similarity between any two documents (e.g., an incoming query and a stored document) is now computed as a scalar product of size K, which is potentiatlly much smaller than r and V.

In the LSA model, the choice of the dimension K is manual and leads to a tradeoff between the precision of the approximation and the desired computational complexity. The higher the value of K, the more faithful is the representation but the higher the computational cost (both in terms of storage and similarity computation).

Another interesting aspect of the LSA model is that each term can also be represented as a vector of size K. In the term-document matrix T_D, the v-th term of the vocabulary is represented by the v-th row $Y^v \in \mathbb{R}^D$. We use the matrix $(\hat{T}_K)^\top$ and similar formulations than above applied to $(T_D)^\top$ to transform Y^v to a K-dimensional vector:

$$(Y^v) = Q\text{diag}(\sigma)(p^v) \approx (Q_K)\text{diag}(\sigma_K)(p_K)^v,$$

where $p^v \in \mathbb{R}^r$ is the v-th row of P and $(p_K)^v \in \mathbb{R}^K$ is the v-th row of P_K. The representation $y^v \in \mathbb{R}^K$ of Y^v is thus given by:

$$y^v = (Q_K)^\top Y^v.$$

This additional term representation is useful for many applications (e.g., find relations between terms such as synonymy or antinomy, term clustering).

The main limitation of the LSA model and its extensions (e.g., the topic-based vector space model [Becker and Kuropka, 2003]) is its underlying assumption that words and documents follow a joint Gaussian distribution (where negative log-likelihood is a squared Frobenius norm). This assumption comes from the fact that the rank-K approximation of the term-document matrix T_D computed with SVD is obtained by minimizing the Frobenius norm $\|T_D - M\|_F$ over all possible rank K matrices M. Thus, as only positive integers are observed in the term-document matrix with word counts representation, Poisson and multinomial distributions are more adapted. For other scores than counts such as TF-IDF, the observed entries are positive and the Gaussian prior is still not appropriate. In the following, we present the probabilistic extensions of LSA for modelling words and documents with Poisson and multinomial distributions.

1.2 PROBABILISTIC TOPIC MODELS

The LSA model [Deerwester et al., 1990] presented above is a dimension reduction model, where we assume the words and documents are generated from at

most K latent features. This constraint is set as a rank constraint on the term-document matrix T_D in LSA and the fitting is done with the Frobenius norm:

$$\hat{T}_K = \begin{cases} \underset{M \in \mathbb{R}^{V \times D}}{\arg\min} & \|T_D - M\|_F \\ \text{s.t.} & \text{rank}(M) = K. \end{cases}$$

As a result, the latent features correspond to the singular vectors of the term-document matrix. Several limitations appear in this model. First, as discussed above, the implicit Gaussian assumption of the LSA model does not match Poisson and multinomial observations. Then, if the number of documents D is large, the SVD decomposition of the term-document matrix may be intractable to compute in practice. Another issue in practice is that if D' new documents are added to the training set, it is very costly to compute the SVD decomposition on the $V \times (D + D')$ matrix and it is not straightforward to include the contribution of the D' new documents to update the latent features (i.e., to update the singular vectors and singular values of the new term-document matrix).

Probabilistic topic models offer solutions to these limitations. The four main characteristics to define such models are listed below:

- As LSA and previously described models, probabilistic topic models are **unsupervised**. This type of models is easily transposable to different datasets as no annotation is required. Manual setting is reduced as much as possible;

- Probabilistic topic models are **latent variable models**. Each observation is described as a mixture of latent features. This property is particularly useful because the number of latent features is usually much smaller than the initial number of dimensions (for instance, in LSA, the initial dimension is the size of the vocabulary V and is reduced to $K \ll V$ in practice);

- Most of topic models are **generative**. Documents are generated from a mixture of latent components and any previously unseen (or test) document can still be described with the latent components learned from the training documents. In this context, the best model will be the one which has the best generative power, i.e., capable of accurately describe any unseen document coming from the same distribution—in practice, the same dataset—than the training documents;

- As induced by the name, the generative process is based on particular **probability distributions**. They offer flexibility of modelling. As mentionned above, they offer a solution to the limitations of LSA and the Gaussian modelling assumption.

The LSA model falls in the first two categories (unsupervised and latent variable model) but we cannot generate new documents from the learned parameters (singular vectors and singular values in LSA) and the learned paramaters are

not probability distributions parameters, even if we can prove that this model best fits observations drawn from a Gaussian distributions.

Mixture of unigrams [Nigam et al., 2000]

The first model example in this context is the *mixture of unigrams* model [Nigam et al., 2000]. The assumption is that each document d is generated by first choosing a mixture component $z \in \{1, \dots, K\}$, called *topic*, then generating the document according to the distribution $p(d|z)$. The probabililty of document d is given by:

$$p(d) = \sum_{z=1}^{K} p(z)p(d|z).$$

In the *mixture of unigrams* model, the naive Bayes assumption is used in order to model $p(d|z)$. Given the topic z of document d, the words of this document are assumed independently distributed. The probability of document $d = (v_1, v_2, \dots, v_{N_d})$ is then, under the naive Bayes assumption:

$$p(d) = \sum_{z=1}^{K} p(z) \prod_{i=1}^{N_d} p(v_i|z).$$

The graphical representation of the *mixture of unigrams* model is presented in Figure 1a. In this particular model, the naive Bayes assumption matches the bag-of-words representation. We denote $X \in \mathbb{R}^V$ the bag-of-words representation with frequency score of document $d = (v_1, \dots, v_{N_d})$, i.e., $X_v = n_v$ for $v \in \{v_i\}_{i=1,\dots,N_d}$ and $X_v = 0$ otherwise, where n_v is the number of occurrences of word v in document d. We then write the probability of document d as follows:

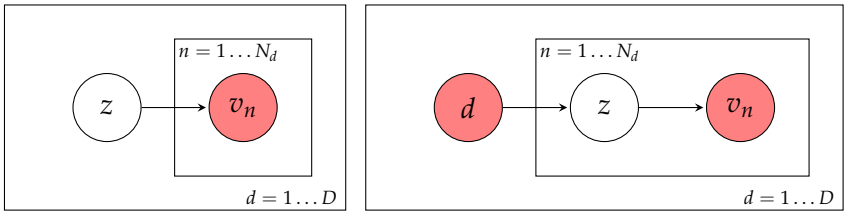
$$p(d) = \sum_{z=1}^{K} p(z)p(X|z),$$

with $p(X|z) = \prod_{v=1}^{V} p(v|z)^{X_v}$. This formulation suggests a multinomial distribution for the conditional $p(X|z)$, which matches the observations. The parameters of the model are the distribution $p(z), z = 1, \dots, K$ and the conditional $p(v|z)$, for $v = 1, \dots, V$ and $z = 1, \dots, K$, which makes $K + VK$ parameters. In practice, the parameters are estimated by maximum likelihood of a corpus $\mathcal{C} = \{d^1, \dots, d^D\}$, where documents are independent:

$$\max_{p(z),p(v|z)} \quad \mathcal{L} = \prod_{d \in \mathcal{C}} p(d).$$

This maximization is done with EM algorithm [Dempster et al., 1977].

As each document is attached to a single topic z, the intuition of this model is that words occurring in the same document are associated to the same topic, represented by the hidden variable $z \in \{1, \dots, K\}$. Each topic z is also associated to the conditional $p(v|z)$ for $v = 1, \dots, V$. We expect that words that have the



(a) Mixture of unigrams. (b) pLSI.

Figure 1: Graphical representation of two probabilistic topic models. White nodes represent hidden variables and colored nodes represent observed variables.

highest probability $p(v|z)$ to appear under the topic z exhibit semantic similarity. In other words, we expect the semantics of topic $z \in \{1, \dots, K\}$ to be conveyed within the conditional $(p(v|z))_{v=1,\dots,V}$.

One limitation of the *mixture of unigrams* model is that in practice we rather expect that a document is generated from a mixture of different topics. For instance, if we consider restaurant reviews, we expect that the topics discussed in the reviews are, for instance, related to the *food*, the *price*, the *service*, the *atmosphere*. It is not correct to assume that all the reviews only tackle one of these topics. We rather have to consider that in all the reviews, the topics are tackled with different proportions for each review. In the restaurant reviews example, some users mostly assess the *food* and the *service* in their review while other users tackle the *food* and the *price* of the meal.

Probabilistic semantic indexing [Hofmann, 1999b]

A first extension of the *mixture of unigrams* model that considers each document is generated from a mixture of topics is the *probabilistic latent semantic indexing* (pLSI) model [Hofmann, 1999b], also known as the *probabilistic latent semantic analysis* (pLSA) model [Hofmann, 1999a]. This model is a probabilistic extension of the LSA model [Deerwester et al., 1990] described above and aims at fixing the (wrong) Gaussian assumption underlying the LSA model. In the pLSI model, the joint probability of word v and document d, namely $p(v, d)$, is parameterized. The underlying generative process with pLSI is the following:

1. Select the document d in a corpus $\mathcal{C} = \{d^1, \dots, d^D\}$ with probability $p(d)$;
2. For each of the N_d words of document d:
   a) Choose a latent topic $z \in \{1, \dots, K\}$ with $p(z|d)$;
   b) Choose a word $v \in \{1, \dots, V\}$ with probability $p(v|z)$.

More formally, the joint probability model on (v, d) is:

$$p(v, d) = p(d)p(v|d), \text{ with } p(v|d) = \sum_{z=1}^{K} p(v|z)p(z|d).$$

Note that in this model, one latent variable (or topic) $z \in \{1, \dots, K\}$ is attached to each word v of document d (i.e., N_d hidden variables in document d), while in the *mixture of unigrams* model only one topic z is attached to each document. Another distinction is the probability $p(z|d)$ which depends on the document d in the pLSI model while $p(z)$ is indepedent to the documents in the *mixture of unigrams* model. The graphical representation of the pLSI model is presented in Figure 1b.

The parameters of the model are the document distribution $p(d)$, the latent variables distribution for each document $p(z|d)$ and the topic distributions $p(v|z)$. In the end the number of parameters is $D + KD + KV = K(D + 1 + V)$. The maximum log-likelihood of the corpus $\mathcal{C} = \{d^1, \dots, d^D\}$ gives an estimate of these parameters:

$$\max_{p(d),p(z|d),p(v|z)} \quad \mathcal{L} = \sum_{d \in \mathcal{C}} \sum_{v \in d} n_v^d \log \left[p(v, d) \right],$$

where n_v^d is the number of occurrences of word v in document d. The EM algorithm is used for this maximization. Note that pLSI is not generative, as the probability of unseen documents $p(d_{new})$ is not known and is estimated for training documents only. The document variable d is only a document index in the corpus. In other words, given the parameters of the model, we can not generate new documents.

We can link the pLSI model to the LSA model with an alternative fomulation of the joint probability $p(v, d)$. We use the identity $p(d)p(z|d) = p(z)p(d|z)$ in the model to compute an equivalent formulation:

$$p(v, d) = \sum_{z=1}^{K} p(z)p(v|z)p(d|z).$$

In this formulation, the occurence of word v is independent from document d given the topic associated to word v: $p(v, d|z) = p(v|z)p(d|z)$. If we denote $\widetilde{T} \in \mathbb{R}^{V \times D}$ the matrix such that $\widetilde{T}_{vd} = p(v, d)$, $\widetilde{P} \in \mathbb{R}^{V \times K}$ the matrix such that $\widetilde{P}_{vz} = p(v|z)$, $\widetilde{Q} \in \mathbb{R}^{D \times K}$ the matrix such that $\widetilde{Q}_{dz} = p(d|z)$ and $\widetilde{\sigma} \in \mathbb{R}^K$ the vector such that $\widetilde{\sigma}_z = p(z)$, the joint probability model $\widetilde{T}$ can be written as a matrix product:

$$\widetilde{T} = \widetilde{P} \mathrm{diag}(\widetilde{\sigma}) \widetilde{Q},$$

which makes a clear link between pLSI and LSA. While in LSA, the K latent factors P^K and Q^K are orthogonal singular vectors, in pLSI the matrices $\widetilde{P}$ and $\widetilde{Q}$ are not orthogonal and correspond to conditional distributions of the model. The main difference between LSA and pLSI thus resides in the objective function. The Frobenius norm used in LSA implicitly models the term-document coefficients with a Gaussian distribution. As a result, the approximation T_K of T_D with LSA is not faithful to the observations—integer word counts—and may even contain negative entries. On the contrary, pLSI properly defines a probabilistic model of the term-document co-occurrences based on multinomial sampling. The objective function is the likelihood function of the observations and

we can show that the maximum likelihood computation with pLSI corresponds to the minimization of the Kullback-Leibler divergence between the empirical distribution (T_D) and the model ($\widetilde{T}$):

$$\begin{cases} \underset{p(d),p(z|d),p(v|z)}{\arg\min} & \mathrm{KL}\left(T_D||\widetilde{T}\right) \\ \text{s.t.} & \mathrm{rank}(\widetilde{T}) = K. \end{cases}$$

In practice, the pLSI model is more flexible and better handles polysemy than LSA. As the latent factors are not necessarily orthogonal, pLSI model allows correlation between latent topics $p(v|z)$, which is more realistic. In LSA, the orthogonality of latent topics typically implies that any word v of the vocabulary only falls in a single category, i.e., in the extreme case, $p(v|z^*) \approx 1$ and $p(v|z) \approx 0$ for $z \neq z^*$. However, for polysemous words, this constraints does not hold. For instance, in a movie reviews dataset, we expect that words like *shoot*, *mole* or *model* appear in two different topics. The word *shoot* may be related to the cinematography or the action to fire a gun. The word *mole* may refer to the animal or an infiltrated spy. The word *model* may refer to a miniature object or a mannequin.

The pLSI model can be seen as a nonnegative matrix factorization (NMF) model [Dhillon and Sra, 2005] with specific (non Gaussian) generative assumptions on the term-document matrix. While the multinomial distributions are adapted to documents, the model may be adapted to fit any type of positive data [Paatero and Tapper, 1994, Lee and Seung, 1999].

The main limitations of the pLSI model are the number of parameters which grows linearly with the number of observed documents D and the fact that this model is not generative and has a limited predictive power on unseen documents.

In the next section, we describe the latent Dirichlet allocation model (LDA) model [Blei et al., 2003] which intends to make the best of both *mixture of unigrams* and pLSI models. On the one hand, the *mixture of unigrams* model is generative and has a fixed number of parameters. On the other hand, with pLSI the documents are modelled as mixtures of different topics. The LDA model combines these advantages.

1.3 LDA, EXISTING EXTENSIONS AND APPLICATIONS

The latent Dirichlet allocation (LDA) model [Blei et al., 2003] is a generative extension of the pLSI model. In LDA, the topic distribution $\big(p(v|z)\big)_{v=1,\dots,V}$ is denoted $\beta^z \in \mathbb{R}^V$, with the constraints $\sum_v \beta_v^z = 1$. The difference with pLSI is that the topic proportions vector of document d is a parameter in pLSI (denoted $\big(p(z|d)\big)_{z=1,\dots,K}$ above), while in LDA the topic proportions vector is a random variable, denoted θ, set with a Dirichlet prior. More formally, document d is generated as follows:

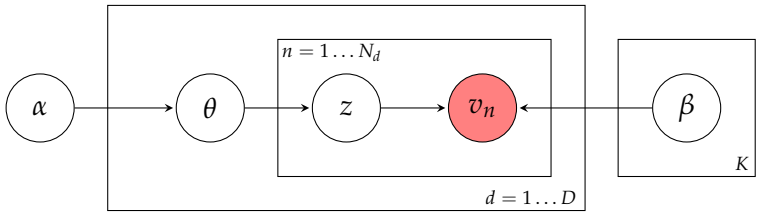


Figure 2: Graphical representation of LDA. White nodes represent hidden variables and colored nodes represent observed variables.

1. Draw the topic proportions θ of document d, with $\theta \sim \text{Dirichlet}(\alpha)$;
2. For each of the N_d words of document d:
   a) Choose a latent topic $z \sim \text{Multinomial}(\theta)$;
   b) Choose a word $v \sim \text{Multinomial}(\beta^z)$.

The parameters of the model are the topic matrix $\beta \in \mathbb{R}^{V \times K}$ and the Dirichlet prior $\alpha \in \mathbb{R}^K$ on the document topic proportions, that is $KV + K$ parameters. The probability of document d generated with LDA is given by:

$$p(d|\alpha, \beta) = \int_\theta p(\theta|\alpha) \prod_{v \in d} \left(\sum_{z=1}^{K} p(z|\theta) p(v|z, \beta) \right) \mathrm{d}\theta.$$

This probability is intractable to compute in practice, so the EM algorithm can not be applied directly to the LDA model. The maximum likelihood is instead estimated with variational EM [Blei et al., 2003, Hoffman et al., 2013], Gibbs sampling [Griffiths and Steyvers, 2002] or moment matching [Podosinnikova et al., 2015]. We tackle and propose a new online inference scheme (i.e., inference from streams of documents) for intractable latent variable models in Chapter 3.

As the previous presented models, the generative process of LDA can be seen as a factorization on the bag-of-words representation $X \in \mathbb{R}^V$ of document d, considered as a random variable. In the LSA model, the underlying generative process is:

$$X \sim \text{Gaussian}(Px, I_V), \tag{LSA}$$

where $P \in \mathbb{R}^{V \times K}$ is an orthogonal matrix and $x \in \mathbb{R}^K$ is a low-dimension representation of X. In the *mixture of unigrams* model (MU), each document is generated from a single feature z:

$$X|z \sim \text{Multinomial}(\widehat{P}^z, N_d), \tag{MU}$$

where $\widehat{P}^z \in \mathbb{R}^V$ is a topic vector $\widehat{P}^z_v = p(v|z)$. In the pLSI model, each document d has its own factorization model:

$$X|d \sim \text{Multinomial}(\widetilde{P}\widetilde{x}, N_d), \tag{pLSI}$$

where $\widetilde{P} \in \mathbb{R}^{V \times K}$ is a topic matrix $\widetilde{P}_{vz} = p(v|z)$ and $\widetilde{x} \in \mathbb{R}^K$ is a vector of joint distribution $\widetilde{x}_z = p(d|z)p(z) = p(d,z)$. Finally, in LDA, given the topic proportions θ, the generative process is independent of the document. In other words, the distribution $X|\theta$ does not depend on the document and is factorized as:

$$X|\theta \sim \text{Multinomial}(\beta\theta, N_d), \tag{LDA}$$

where $\beta \in \mathbb{R}^{V \times K}$ is a topic matrix $\beta_v^z = p(v|z)$ [Buntine and Jakulin, 2005].

In LDA, each document is modelled with a mixture of different topics and the number of parameters does not depend on the size of the corpus. In particular, online inference—i.e., learn the parameters from streams of data—is much more practical than with the other presented methods. The LDA model is easily enhanced by adding new variables (see next section). Even if the resulting generative process is much more sophisticated, the inference scheme for these extensions is usually very similar to the LDA inference, which explains the attractiveness of this model. As a matter of fact, the LDA model is widely popular (almost 20,000 citations; pLSI has 4000 citations), has been used for many applications and has led to many extensions.

1.3.1 Extensions of LDA

There exist many extensions of LDA to match particular applications or to correct a specific limitation of the model.

Topic coherence improvement.

In the LDA model, when the number of topics K is very large, some topics are "junk" topics, namely topics that are not consistent around an aspect (e.g., *food* or *service* in a restaurant reviews dataset). A solution to this issue is to automatically filter out these "junk" topics by ranking them [AlSumait et al., 2009]. Another solution is to regularize the objective function [Newman et al., 2011]. The empirical judgement of topics by users is discussed by Chang et al. [2009].

Topic structure.

The first extensions of LDA found in the litterature are related to the structure of the topic matrix. In LDA, the topics are represented by $\beta \in \mathbb{R}^{V \times K}$ as a list of K independent discrete distributions on the vocabulary of size V, where the number of topics K is set manually. One of the limitations of LDA is that it does not model the correlation between topics although we expect that a document about *politics* is more likely to also tackle *elections* than *sport*. In the correlated topic model (CTM) [Blei and Lafferty, 2007] the Dirichlet prior on θ in LDA (the topic proportions of the document) is replaced by a logistic normal distribution to parameterize and learn the correlation between topics. As a result, on top of

learning a topic matrix β, explicit links between the K topics are learned and we can represent the topics as a graph where an edge between two topics is set if their correlation is high enough (see Blei and Lafferty [2007] for more details and examples of such graphs). A related model is the latent topic network model (LTN) [Foulds et al., 2015], where the Dirichlet prior on θ is replaced by a conditional random field (CRF). A graph of topics is thus learned with this model. Another approach to model the correlation of topics is to put a prior on the topic matrix β, parameterized as a sum of regression factors [Paul and Dredze, 2012].

Non-parametric approaches.

Another limitation of the LDA model is the number of topics K which has to be set manually. Several non-parametric extensions of LDA adaptively learn the number of topics from the documents. For instance, in the hierarchical Dirichlet process (HDP) [Teh et al., 2006], a list of topics is iteratively updated by adding or removing topics of the list. More complex structures than lists are also used to model the topics. Both the nested Chinese restaurant process (nCRP) [Blei et al., 2010] and the nested hierarchical Dirichlet process (nHDP) [Paisley et al., 2014] learn a (non-parametric) tree of topics from documents. In this tree, the root topic is the most generic while a leaf topic is very specific (e.g., *sport* is the parent topic of *soccer* and *baseball*). See an efficient algorithm for such models in Chapter 3.

Multi-scale topics.

The LDA model does not consider the structure of the documents, i.e., the sentences and the order of words. In the multi-grain LDA (MG-LDA) model [Titov and McDonald, 2008], two scales of topics are learned. A list of "global" topics describing a whole document and a list of "local" topics describing sentences. Consequently, this models takes into account the structure of the documents while still keeping the practical bag-of-words representation for the documents and sentences.

Dynamic topic models.

In the LDA model, the temporal order of documents is not considered. However, for large dynamic datasets such as news, the topics discussed in the news ten years ago are probably strongly different than the topics of nowadays news. In the dynamic topic model (DTM) [Blei and Lafferty, 2006], a discrete evolution of the topic matrix is considered by grouping documents in temporal bins. The continuous time extension of the DTM model, called continuous time dynamic topic model (cDTM) [Wang et al., 2012], in which the timestamp of each document is considered. Other approaches to model the temporal evolution of text datasets represent words co-occurences as a dynamic graph over time [Palla et al., 2016].

Opinion mining.

In the ASUM model [Jo and Oh, 2011], a hidden "sentiment" variable is attached to each sentence and is used to extract positive and negative topics. In the topic sentiment mixture model [Mei et al., 2007], each of the K latent features is divided in three topics: one neutral, one positive and one negative topic. In the restaurant reviews example, each aspect—*food*, *service*, *atmosphere*, *price*—would be attached to three topics that respectively convey neutral, positive and negative words about this aspect. In this model, the underlying generative process of a document is the following: we first choose one of the K features, then one of the three sentiment (expected to be neutral, positive or negative), finally we sample a word from the corresponding topic (e.g., topic of positive word distribution about *food*). These three steps are performed by sampling from multinomial distributions. In the joint sentiment/topic model (JST) [Lin and He, 2009], S lists of K topics are learned, each list corresponding to a sentiment (typically, $S = 3$ to model K neutral, K positive and K negative topics). The generative process for words is the following: one of the sentiment features $s \in \{1, \dots, S\}$ is choosen, then one of the K topics of list s is sampled and finally a word from this topic is drawn. Note that in the JST model, there is not necessarily correspondance between neutral, positive and negative topics. It may happen that the *price* aspect only appears in the neutral list of topics while the *food* aspect appears in the positive and negative list of topics (but not in the neutral list).

Recommendation.

In crowd-sourced review services such as Yelp, IMDB, Amazon etc., each user assesses a particular content (e.g., a movie, a restaurant, a product) with a numerical rating together with a detailed text comment of her opinion on the different aspects of the content. A recommender system is built by combining ratings and text reviews in LDA. In the supervised LDA model [Mcauliffe and Blei, 2008], the rating is parameterized as a response to the corresponding text review. As a result, given an unseen text review, the model is able to predict the rating attached to this review. Each topic learned with this model is attached to a score which represents the influence of this topic on the final rating. Another solution to combine rating and text in LDA given by McAuley and Leskovec [2013] is to explicitly parameterize the topic proportions of the review θ as a function of the rating r attached to the review. We propose in Chapter 2 an extension of LDA to combine ratings and text reviews in which we distinguish the topics that convey opinion from descriptive (neutral) topics.

1.3.2 Limitations

In a recommender system where we want to suggest contents together with personalized text, the main limitation of the presented topic models is the bag-of-words representation. Indeed, with this representation, we can only suggest

bag-of-words documents to user, i.e., lists of single words. For instance, if you are suggested the movie Batman, a topic model like LDA would probably attach the words *batman*, *action*, *good*, *movie*, *robin*. The issue is that this list of words may not be intuitive to make a decision about the content. It would be more intuitive to have readable sentences such as *Batman is a good action movie*. This requires to represent differently the documents and in particular to model the dependency between words of a document. In the next section, we discuss other approaches to represent and model documents; see also Chapter 4.

1.4 BEYOND LDA AND BAG-OF-WORDS REPRESENTATION

As explained above, the bag-of-words representation for documents is practical but totally dismisses the structure of a document. In particular, both the order of words and sentences cutting are ignored and words are assumed to be (conditionnally) independent, which is an issue. Indeed, we expect that, if we are given the word v_n in the n-th position of a document, the distribution of the preceding word $p(v_{n-1}|v_n)$ is different than the distribution of the following word $p(v_{n+1}|v_n)$. For instance, in a restaurant review, if we know that $v_n = hot$, it is more likely that the preceding word is *good* and the following word is *dog* than the other way around (*good hot dog* is more likely to appear than *dog hot good*). More formally:

$$p(v_{n+1} = dog|v_n = hot) \gg p(v_{n+1} = good|v_n = hot),$$
$$p(v_{n-1} = dog|v_n = hot) \ll p(v_{n-1} = good|v_n = hot).$$

In LDA and other topic models, this order relation is ignored.

A n-gram is defined as a sequence of n consecutive words $(v_1, v_2, \ldots, v_n)$. The order is important here as for instance $(v_1, v_2) \neq (v_2, v_1)$ for $n = 2$. Note that in a bag-of-words representation, the pairs (v_1, v_2) and (v_2, v_1) are strictly equivalent. The n-gram class model [Brown et al., 1992] addresses the problem of predicting a word given the $n-1$ previous words in a sample of text. In this model, instead of directly estimating the distributions $p(v|v_1, v_2, \ldots, v_{n-1})$ which require $V^n - 1$ parameters, a class $c \in \{1, \ldots, C\}$ is associated to each words and the model follows the assumption:

$$p(v|v_1, v_2, \ldots, v_{n-1}) = p(v|c_n)p(c_n|c_1, \ldots, c_{n-1}),$$

where c_i is the class of the word in position i in the sequence. This assumption follows the intuition that the vocabulary may be clustered in C classes of words and words in the same class are interchangeable in a sample of text. For instance, if we have to finish the sentence *"My appointment is on ___"*, the words *Monday* and *Thursday* are equally probable to occur. Therefore, they would be assigned the same class and the number of parameters to fill the sentence is reduced from V to C. If the size of the vocabulary V is large, this modelling assumption reduces significantly the number of parameters involved. Brown et al. [1992]

propose algorithms to approximate the maximum likelihood of the 2-gram class model.

However, the n-gram models are not generative as their goal is to predict the last word of a n-gram given the $n-1$ previous words. The bigram topic model [Wallach, 2006] combines the LDA model with a 2-gram (or bigram) model. In this model, the word v_t in position t is generated from the previous word v_{t-1} and the chosen mixture for this word $z_t \in \{1, \dots, K\}$. The context v_{t-1} has an influence on the generative process. The parameters learned are the distributions $p(v_t|v_{t-1}, z_t)$ which correspond to $V \times K$ topics (instead of K topics in LDA). While the choice of topic z_t for word t is the same than for LDA $p(z_t|\theta)$, each word is now generated from the topic matrix $\phi \in \mathbb{R}^{VK \times V}$, with $\phi_v^{(j,k)} = p(v_t = v|v_{t-1} = j, z_t = k)$. The combination of the LDA model with a bigram model results in a generative model that takes the structure of the document (or the context) into consideration. If we extend this model to n-grams with $n > 2$, namely model the distributions $p(v_t|z_t, v_{t-1}, v_{t-2}, \dots, v_{t-n+1})$, the number of parameters required grows exponentially with the length of the n-gram $O(KV^n)$ with no further assumptions, which becomes computationally intractable for large vocabulary V.

Other scalable approaches have been developped to model long contexts. One recently popular approach use *word embeddings*. The princple is to map each word v of the vocabulary in a r-dimensional space. Given a corpus $\mathcal{C}$ of observed documents, the vectors $(X^v \in \mathbb{R}^r)_{v=1,\dots,V}$ associated to words of the vocabulary are learned from $\mathcal{C}$ by minimizing a cost function (or maximizing a likelihood) on the embeddings. This cost function is typically set so that words that are used in similar context (i.e., interchangeable words) have similar embeddings.

This *word embeddings* approach has recently gain popularity with the skip-gram (or word2vec) model [Mikolov et al., 2013b]. In this model, the online inference of the embeddings is efficiently implemented and requires absolutely no supervision. The principle is to maximize the likelihood of words in a window:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(v_{t+j}|v_t).$$

The window $(v_{t-c}, \dots, v_{t-1}, v_{t+1}, \dots, v_{t+c})$ is the context of word v_t and the window size c is manually set in practice. In the skip-gram model, the probability $p(v_{t+j}|v_t)$, for $-c \leq j \leq c$, is parameterized as a soft-max function of the embeddings:

$$p(v_{t+j}|v_t) = \frac{\exp\left(\langle X^{v_{t+j}}, X^{v_t} \rangle\right)}{\sum_{w=1}^{V} \exp\left(\langle X^w, X^{v_t} \rangle\right)}.$$

This raw skip-gram formulation is impractical as the computational cost of the gradient $\nabla \log p(v_{t+j}|v_t)$ is proportional to the vocabulary size V, which is large for real datasets ($10^4 - 10^7$). Along with this skip-gram model, Mikolov et al. [2013b] combine different tricks to efficiently maximize the likelihood $\mathcal{L}$. Conse-

quently, this model and other related *word embeddings* models have been used for many applications as word vectors are quickly computed and the comparison of two words is efficient. For instance, this type of models covers question answering [Mikolov et al., 2013a, Bordes et al., 2014], document classification [Kusner et al., 2015, Huang et al., 2016], automatic translation [Zou et al., 2013].

Even if the *word embeddings* models are powerful tools to effectively parameterize the structure of documents, they only provide seperate representations for words of the vocabulary. Given the vectors associated to every words of the vocabulary, it is not straightforward to compute the representation of a full sentence or a full document as a vector. We propose in Chapter 4 a document model based on determinantal point processes (DPP). In our model, a document is represented as a list of sentences and each document is generated by selecting a subset of all the possible sentences. The probability of a subset is parameterized with a DPP. Note that with our representation, the sentences of a document are *not* independent. We apply our model to document summarization where we propose readable summaries of restaurant reviews.

1.5 CONTRIBUTIONS

In this thesis, given the models previously described, we make the following contributions to the problem of suggesting personalized annotated content to the users:

- Chapter 2. We first observe that topic models like LDA mostly extract descriptive words and give a very low importance to words that convey opinion. We propose a model that leverages ratings together with reviews to distinguish qualitative and descriptive words. While several existing methods include ratings in the LDA model (e.g., Mcauliffe and Blei [2008], McAuley and Leskovec [2013]), their goal is to improve the rating prediction by adding the text information. Moreover, topics that convey opinion are not clearly identifiable with such methods. As we want to suggest text to the user, our method rather leverages ratings to improve review prediction. We run and evaluate our method on a movie reviews dataset. Not only does our model improve the overall predictive power compared to existing work but it also extracts meaningful positive and negative topics that are automatically identifiable. This work is accepted at the 13th International Conference on Machine Learning and Data Mining (MLDM) 2017.

- Chapter 3. For very large corpora or online review services, it is convenient nay necessary to process the dataset little by little. We provide a method to efficiently learn parameters from streams of data when the model posteriors are intractable to compute (which is usually the case for topic models). There exist many online inference schemes for the LDA model but the link between these methods is not explicit. We propose a unifying frame-

work to draw explicit links between these methods and propose a new online method in this framework that outperforms the previously proposed schemes. This work is under revision at JMLR.

- Chapter 4. Determinantal point processes (DPPs) are useful to model diversity for subset selection problems. These models suffer from a high computational cost when the number of items to choose from is large. We propose a new class of DPPs based on a specific low-rank factorization of the marginal kernel, which is particularly suited to DPPs defined on exponentially many items. We apply this new class to modelling text documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions, which is made possible with no approximation for our class of DPPs. More precisely, given the topic proportions, a document is generated with our model by selecting a subset of diverse sentences among all the possible sentences and the probability distribution over the subsets of sentences (i.e., over the possible documents) is formulated as a DPP. We present an application to document summarization on a restaurant reviews dataset with a DPP on 2^{500} items. We observe that our model is able to generate meaningful summaries and extract word embeddings and topics that convey semantics, while keeping a reasonable computational complexity. This work is submitted to ICML 2017.

2

PERSONALIZED REVIEW PREDICTION WITH LATENT DIRICHLET ALLOCATION

Crowdsourced review services allow any user to assess contents (e.g., movies, restaurants, hotels) through both numerical rating and free form text. They exploit the ratings to suggest contents to users while making raw reviews available for users to get a precise opinion on a particular content. As it is tedious to extract useful information among thousands of reviews, it is more intuitive and less time consuming for users to access a personalized summary of the other users' opinions on each content. In this regard, many existing implementations either manually label each content—e.g., with the genre—or select "useful" reviews to read in order to make a quick opinion on the particular content.

These existing approaches suffer from several limitations. First, the textual information—labels of contents or "useful" reviews—provided by the system is not personalized and may not be adapted to every user. The aspects decribed in the selected "useful" reviews may not be decisive in the opinion of every user and the labels of contents may be too generic to make a decision. The labelling of the movies is also a cumbersome human task.

Consequently, our goal is to suggest the user a personalized text summary of what could be her opinion on each content she visits, in order to enhance recommender systems.

In our approach, we combine both text reviews and numerical ratings to automatically predict the words a user would employ to assess an unseen content. We leverage topic models to separate descriptive information and qualitative information in distinct topics. This distinction between descriptive and qualitative information is crucial as it enables to decompose the reviews in (1) the aspects evaluated in the reviews and (2) the opinion attached to these aspects. Without this distinction, it is impossible to automatically identify the decisive aspects in the opinion of the user. We extend the LDA (latent Dirichlet allocation) model [Blei et al., 2003] to include ratings along with texts.

Our contributions are (1) extensions of LDA in order to separate descriptive and qualitative topics in a corpus of text reviews using movie ratings and (2) their theoretical evaluation and comparison to state of the art [McAuley and Leskovec, 2013, Mcauliffe and Blei, 2008] for the task of word prediction. As a by-product (3) we show that profiling users from reviews is not possible with this method because of the diversity of movies reviewed and opinions expressed by users. Note that this work can easily be generalized to other types of content such as restaurants, hotels and products. We have indeed applied our models on a restaurant dataset with similar observations and performance.

This work will be published in the proceedings of the 13th International Conference on Machine Learning and Data Mining (MLDM) 2017.

2.1 TOPIC EXTRACTION WITH LDA

We define a corpus as a list of documents and a document (e.g., a single user review) as a list of words. In this section we present the methodology we use to pre-process raw text reviews. We then describe two types of topic models. The first type of model only applies to text while the second type of model applies to text and ratings. Each review is generated by a user on a movie. We apply each type of model to three different corpora, depending on wether a document is (1) a single review, (2) the concatenation of reviews related to a single movie or (3) the concatenation of reviews written by a single user. These three different processes with the two types of models lead us to six different models.

2.1.1 Pre-processing Reviews

Many words in reviews are not relevant for our purpose as they convey neither qualitative nor descriptive information (e.g., stop words), or because they appear too frequently in reviews and are too generic (e.g., *movie*, *film*, *scene* in movie reviews).

We first remove the stop words using the NLTK toolbox [Bird et al., 2009] and words appearing in more than 20% of the reviews (for instance, *movie* appears in 80% of the reviews). We choose 20% as a threshold because it only filters out very frequent words. We then select from the remaining words the 10,000 most frequent words in the database. We observe that in our IMDB dataset of 97,000 reviews (we use a subset of the dataset decribed by Diao et al. [2014] that spreads over 5,900 movies and 2,400 users), after filtering, each word appears in at least 10 reviews, which means only words that appear in less than 10 reviews—i.e., 0.01% of the dataset—have been pruned.

2.1.2 LDA and Extensions

Let D be the number of documents of a corpus $\mathcal{W} = \{\mathbf{w}^1, \ldots, \mathbf{w}^D\}$, $V = 10,000$ the number of words in the vocabulary and K the number of latent topics in the corpus. Each topic ϕ^k corresponds to a distribution on the V words. For each document d, LDA infers a discrete distribution θ^d over the K topics. In practice, the inference may be done using variational EM [Blei et al., 2003], Gibbs sampling [Griffiths and Steyvers, 2004] algorithm or online learning [Hoffman et al., 2010]. As detailed in Chapter 3, LDA is a generative model applied to a corpus of text documents which assumes that the n-th word w_n^d of the d-th document is generated as follows:

- Choose $\theta^d \sim \text{Dirichlet}(\alpha)$,

LDA-O, 1 doc = 1 review					LDA-C, 1 doc = 1 movie				
comedy	best	action	horror	love	comedy	horror	action	love	animation
funny	great	comic	thriller	family	funny	original	die	son	animated
laugh	love	great	best	romantic	laugh	remake	hard	best	toy
fun	old	best	killer	comedy	joke	gore	bad	mother	voice
love	totally	fun	great	beautiful	fun	scary	hero	comedy	pixar
great	award	x-man	kill	best	hilarious	dead	john	beautiful	child
best	funny	kill	bad	young	great	scare	fight	young	great
hilarious	benjamin	fight	dark	romance	sex	scream	bruce	brother	disney

Table 1: Top 5 topics extracted with LDA-O and LDA-C, $K = 128$ (ordered by importance $\hat{\theta}_k = \frac{1}{D}\sum_d \theta_k^d$).

- For each word $w_n^d \in \mathbf{w}^d$:
  - Choose a topic $z_n^d \sim \text{Mult}(\theta^d)$,
  - Choose a word $w_n^d \sim \text{Mult}(\phi^{z_n^d})$.

In a recommendation setting, each review refers to a unique (user, movie) pair. One could use this information to learn specific topics of (movies, users) pairs and get a more precise representation of the reviews. In our baseline model, noted LDA-O, we run LDA on a corpus where each document is a single review (i.e., we consider that reviews are independently generated).

We then aggregate our reviews accounting for the fact that multiple reviews either belong to a movie or to a user. We build user profiles in LDA by constraining the topic document distribution θ to be the same for all the reviews written by the same user. It is equivalent to considering a new corpus where each document is the concatenation of all the reviews written by a single user. We refer to this model as LDA-U and we refer to the topic document distributions θ^u inferred with LDA-U as "topic user distributions", where $u \in \{1, \ldots, N_u\}$ denotes users. The same aggregation is done to profile movies with the corpus where a document is the aggregation of all the reviews belonging to the same movie. We refer to this model as LDA-C and we refer to the topic document distributions θ^c as "topic movie distributions", where $c \in \{1, \ldots, N_c\}$ denotes movies.

Inferred Topics.

Topic consistency is an empirical notion that we use in the rest of the chapter. A strongly consistent topic has all its top words belonging to the same lexical field. This notion can be applied to any list of words such as a document or a corpus.

Parameter K influences the consistency of the descriptive topics. Ideally, we would like each topic to represent a movie *feature*, e.g. actors, genres, or sequels.

After multiple experiments with values of K between 8 and 260, we observed that for $K \leq 30$, the descriptive topics mix several features. On the contrary,

for $K \geq 150$, each feature is spread over multiple topics and topic consistency decreases. In Table 1, $K = 128$ is a good compromise [1].

The first 5 topics inferred with the models LDA-O and LDA-C proposed above are presented in Table 1. As the topics extracted with LDA-U are not consistent, they are not presented. We observe that topics in LDA-O and LDA-C are very consistent around a genre, an actor, a director or a sequel. The main difference between these two methods is that there are more outliers in LDA-O topics than in LDA-C topics. These outliers are generic words appearing frequently enough in the corpus to influence the topics. These words are not frequent enough to be removed during the filtering (e.g., *best*, *action*, *review*). In LDA-C, the aggregation of reviews related to the same movie lowers the impact of these outliers.

The corpus used in LDA-C is consistent because reviews of the same movie share a common vocabulary, each document of this corpus brings out words associated to the movie and lowers the influence of noisy words in the topics. We also observe that topics obtained with our models are mostly descriptive and that very few qualitative information is "lost" in the middle of the topics, making it difficult to tell a user whether or not she would like a movie.

2.1.3 Inclusion of Ratings in the Inference Process

Given the lack of qualitative in the top words of the topics, we introduce a method to extract qualitative topics (i.e., words with a positive or negative connotation) using numerical ratings in addition to text reviews. Given a text review and the corresponding rating, we expect user's opinion to be conveyed in the text and summarized in the rating. Ideally, positive (resp. negative) words should be more likely to appear in a high (resp. low) rated review. In order to keep the model simple, we reduce first numerical ratings (initially between 1 and 10) to binary ratings $\{-1, +1\}$. Using these new ratings, we infer a positive (resp. negative) topic from $+1$ (resp. -1) reviews in a new generative model.

Rating Reduction.

We use a standard rating prediction technique [Koren et al., 2009] to extract users and movie features given ratings. We denote by D the number of reviews, r^d the rating of the d-th review, related to user $u^d \in \{1, \dots, N_u\}$ and movie $c^d \in \{1, \dots, N_c\}$ with $d \in \{1, \dots, D\}$. We first model the ratings to be generated as a sum of a user factor and a movie factor (ANOVA, Casella and Berger [2002]):

$$r^d = \mu + a_{u^d} + b_{c^d} + \epsilon, \quad d \in \{1, \dots, D\},$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We optimize (a_u), (b_c) and μ as the solution of a penalized (in L^2 norm) least-squares problem.

[1] For restaurants, the number of features is smaller and the optimal value of K is around 60.

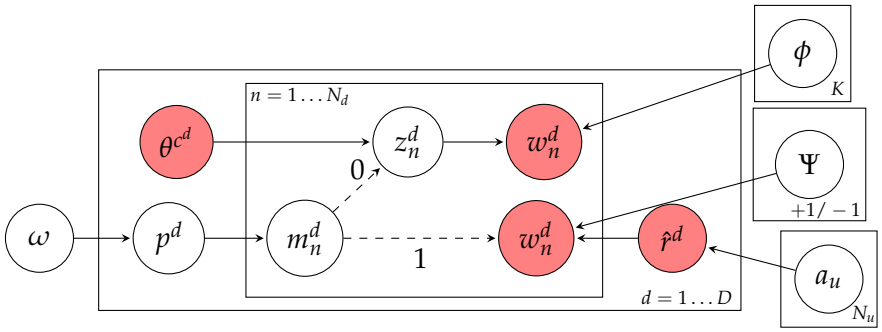


Figure 3: Graphical representation of the model `LDA-R-C` including reduced ratings applied on D documents. White nodes represent hidden variables and colored nodes represent observed variables. The observed rating r^d is not reported for the sake of clarity.

The training ratings are reduced to -1 if the following residual is negative and to $+1$ if the residual is positive:

$$\hat{r}^d = \begin{cases} +1 & \text{if } r^d - (a_{u^d} + \mu) \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

In the model, μ represents the average rating, b_c the average deviation from μ of ratings on movie c, a_u the average deviation from μ of ratings from user u. We consider here that a review is positive ($\hat{r}^d = +1$) if user u^d rated movie c^d with a higher score than her average rating $(a_{u^d} + \mu)$. The reduced rating $\hat{r}^d$ represents the binary user opinion.

For any test review t, the user opinion $\hat{r}^t$ is unknown. This quantity is then a random variable with:

$$\begin{aligned} \mathbf{Pr}[\hat{r}^t = +1] &= \mathbf{Pr}(b_{c^t} + \epsilon \geq 0 | a, b) \\ \mathbf{Pr}[\hat{r}^t = -1] &= 1 - \mathbf{Pr}[\hat{r}^t = +1]. \end{aligned}$$

During prediction of the review t, we use these probabilities to extract a mixture of positive and negative words.

Qualitative Topics Inference

We use these reduced ratings $(\hat{r}^d)$ to infer qualitative topics. We consider a corpus where each document is a single review. We denote by D the number of documents, by $c^d \in \{1, \dots, N_c\}$ and $\hat{r}^d \in \{-1, +1\}$ respectively the movie and reduced rating of the d-th document, $d = 1, \dots, D$. K is the number of descriptive topics, i.e., the number of topics extracted with `LDA-C`. Given K, the corresponding topic movie distributions $\theta^c, c \in \{1, \dots, N_c\}$ inferred with `LDA-C` on the D reviews and the reduced ratings $(\hat{r}^d)$, we extract a positive topic and a negative topic with the following generative model. For each word w_n^d of the d-th document $\mathbf{w}^d$:

- Draw $m_n^d \sim \text{Bernoulli}(p^d)$

- If m_n^d is a success ($m_n^d = 1$):
  - Choose $w_n^d \sim \text{Multinomial}(\Psi^{\hat{r}^d})$, with $\hat{r}^d \in \{+1, -1\}$
- Otherwise ($m_n^d = 0$):
  - Choose a topic $z_n^d \sim \text{Multinomial}(\theta^{c^d})$
  - Choose $w_n^d \sim \text{Multinomial}(\phi^{z_n^d})$.

In other words, if m_n^d is a success, w_n^d is qualitative otherwise w_n^d is descriptive. We still infer K descriptive topics with this method, represented by $\{\phi^1, \ldots, \phi^K\}$. The main difference with LDA-C is that we only infer ϕ, Ψ parameters, knowing topic document distributions θ. We also learn the proportion of qualitative/descriptive words in each document p^d, embedded with a Dirichlet prior. In the following, we refer to this model as LDA-R-C. The graphical representation of LDA-R-C is presented Figure 3.

We build two additional models by replacing in LDA-R-C the topic movie distributions θ^c by the topic document distributions $\theta^d, d = 1 \ldots D$—model LDA-R—or by the topic user distributions $\theta^u, u = 1 \ldots N_u$—model LDA-R-U.

For the three models, the inference is done with a variational EM algorithm adapted from LDA inference [Blei et al., 2003]—see Appendix 2.A for complete derivations. We also run a full EM for learning at the same time both topics (parameters ϕ, Ψ) and topic distribution (parameter θ). As the algorithm is very sensitive to initialization, we observe that random initialization gives poor results while initialization with parameters resulting from LDA-O leads to a steady state. Indeed, LDA-O returns a local miminum of the full algorithm.

Inferred Topics.

As the qualitative topics extracted with LDA-R-U are not consistent, they are not presented. The qualitative topics inferred with LDA-R and LDA-R-C are presented in Table 2. In both LDA-R and LDA-R-C the average proportion $\frac{1}{D}\sum_d p^d$ of qualitative words in documents is 10%. Some qualitative words appear in the top words of the opposed topic (e.g., *bad* comes up in the positive topic of LDA-R). In our model, each training document d is assigned to only one qualitative topic—Ψ^{+1} or Ψ^{-1}—depending on the corresponding rating $\hat{r}^d$. Consequently, if a positive word frequently appears in low rated reviews—e.g., in negation phrases—it is likely to be a top negative word and vice versa. Some neutral words also come up in these qualitative topics (e.g., *suppose*). These words appear in reviews of different types of movies and are then in the tail of descriptive topics. They are still used frequently enough to have an influence on the qualitative topics.

2.2 EVALUATION

In this section we evaluate our model for the review prediction task. Using our models, a review prediction is a discrete distribution over the V words of the

LDA-R		LDA-R-C	
great	bad	great	bad
best	worst	bad	great
nice	boring	love	boring
bad	waste	best	worst
entertaining	stupid	original	stupid
original	terrible	entertaining	totally
love	suppose	definitely	waste

Table 2: Positive (left column) and negative (right column) topic inferred with LDA-R and LDA-R-C, $K = 32$ topics.

vocabulary given a set of training reviews $\mathcal{W}$ and a test couple (u^t, c^t) of user, movie. The evaluation of such prediction is done by splitting the review dataset in a training set and a testing set. The parameters of the model are learned on the training set and evaluated on the test set.

Methodology.

Given a test review w, the best predictor maximizes the likelihood $P(w|\mathcal{W})$ of the test review, P depending on the model. We then use the log-perplexity measure defined by $LP = -\log P(w|\mathcal{W})$ to evaluate our models. The log-perplexity is a theoretical measure of the quality of the model for the word prediction task; it is not an indicator of user satisfaction. As shown by Chang et al. [2009], the perplexity is not suited to measure user satisfaction. However, perplexity measures the precision of the prediction, which is what we need in order to compare our models to the state of the art.

For LDA-O, the likelihood is intractable to compute. We approximate $P(w|\mathcal{W})$ with the "left-to-right" evaluation algorithm [Wallach et al., 2009] applied to each test document. This algorithm is a mix of particle filtering and Gibbs sampling, easily adjustable to other graphical models. For LDA-C, as the topic distributions are learned for each movie (resp. user), the likelihood of a new review is computed for movies (resp. users) seen in the training corpus through pointwise estimation. Finally, we adapt the "left-to-right" algorithm to approximate the likelihood of LDA-R and LDA-R-C for each test document.

We compare our models to two existing approaches. In the model proposed by McAuley and Leskovec [2013], authors use words and ratings to predict ratings by learning a mapping function between LDA parameters and rating matrix factorization parameters. We refer to this model as HFT (which stands for "hidden factors as topics"). In the model proposed by Mcauliffe and Blei [2008], authors incorporate scores directly in LDA in order to predict the score of a new review given the words used. We refer to this model as SLDA. These two methods will be discussed in related works. Both HFT and SLDA use documents and ratings associated to each document to infer similar parameters than LDA—i.e., topics

ϕ and topic proportions θ. While the application presented by Mcauliffe and Blei [2008] and McAuley and Leskovec [2013] is rating prediction, we can use the parameters ϕ and θ inferred with these methods to predict words of new reviews.

We could not compare to the method described by Ling et al. [2014] because of the lack of information available to implement the method accurately.

The review dataset was collected from IMDB, with a catalogue of 5,900 movies, 2,400 users and 97,000 reviews—a subset of the data described by Diao et al. [2014]. We apply our models to 10 random splits of train and test sets, representing respectively 90% and 10%. Each movie and each user of a testing set is seen at least once in the corresponding train reviews.

Figure 4 presents the average log-perplexity measures per word on the test sets for `LDA-O`, `LDA-C`, `LDA-R`, `LDA-R-C`, `HFT` and `SLDA`. Given the poor topic consistency observed with `LDA-U` and `LDA-R-U` and their poor performance, we do not display perplexity for these models.

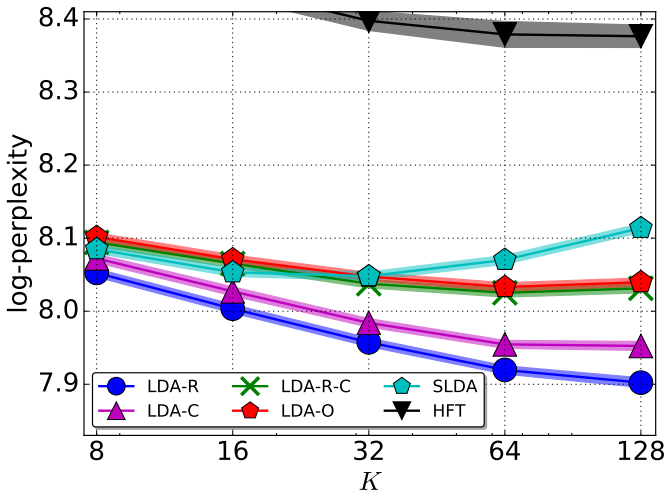


Figure 4: Average log-perplexity per word of the presented models on test reviews.

Evaluation.

The number of topics K is the unique input of LDA and its extensions. It represents the dimension of the latent topic document distribution θ. The Perplexity on test sets still decreases at $K = 128$ for `LDA-C` and `LDA-R`, which means we do not reach overfitting with $K \leq 128$.

`LDA-O` overfits the training reviews as it infers one distribution over topics (θ) by review. This method does not catch the structure of the vocabulary in the reviews as all reviews are processed separately. The predictions with `LDA-O` are then polluted by frequent words which appear in the inferred topics and lower the quality of the predictions. Conversely, the aggregation of reviews in

LDA-C improves the consistency of the topics and generates better movie profiles (through topics movie distribution θ^c). LDA-C outperforms LDA-O.

By comparing the performance of LDA-O and LDA-R, we can observe that including reduced ratings in LDA-O improves perplexity for all values of K. Indeed, for all K, descriptive topics of LDA-R contain noisy words. The representation of the qualitative words in the training set results in an important decrease of perplexity compared with LDA-O.

The consistency of qualitative topics in LDA-R makes predictions outperform LDA-R-C. Predictions with LDA-R-C are also beaten by predictions with LDA-C. The consistency of descriptive topics of LDA-C is affected when including ratings.

With HFT the topic-document distribution θ is mapped to matrix factorization parameters. The objective being optimized can be expressed as a sum: f =(rating prediction error)−(corpus likelihood). The quality of word prediction is deteriorated as the objective is not directly the corpus likelihood. As a result, HFT is the worst predictor.

SLDA overfits at $K = 32$ as it infers one distribution over topics (θ) by review. This model includes ratings in LDA to infer a score along each topic. As a result, SLDA extracts descriptive and qualitative topics but they may not be automatically differentiated (score associated with a descriptive topic may be as high as the score associated with a positive/negative topic). SLDA is able to extract qualitative and descriptive information from the reviews but is outperformed by LDA-C and LDA-R for all values of K.

Even if the best predictors are LDA-R and LDA-C, the most practical model is LDA-R-C for three main reasons. Performances of LDA-R-C (in terms of perplexity) are comparable to the best models. This model is the only model combining strongly consistent descriptive topics with consistent qualitative topics. LDA-R suffers from polluted descriptive topics while LDA-C does not infer any qualitative information. The profiles of movies θ^c extracted with LDA-R-C are useful to compute a distance between movies—e.g., with the Kullback-Leibler divergence. As a result, one could recommend movies to a cold start user (i.e., user with an insufficient number of reviews) with LDA-R-C.

2.3 EMPIRICAL DISCUSSION

In this section, we present the 8 most significant topics out of 100 topics inferred with LDA-R-C, SLDA and HFT in Table 3. The comparison of the topics is a premise to the analysis of words prediction. Indeed, for any LDA based algorithm, the probability of occurence of word w is given by $\sum_k \theta_k^\top \phi_w^k$, where θ represents the topic proportions of the current review and ϕ the inferred topic matrix. A topic vector ϕ^k may be seen as the probabilities of occurence of words in a review generated from a single topic i.e., $\theta_k = 1$ and $\forall j \neq k, \theta_j = 0$.

Note that this section is an empirical discussion and that it does not replace a formal evaluation with real users that is planned for future works due to the complexity of the task (methodologically and business-wise).

LDA-R-C, 1 doc = 1 movie							
Qualitative		Descriptive					
Q1	Q2	T1	T2	T3	T4	T5	T6
great	bad	funny	horror	animation	child	alien	action
love	waste	comedy	scary	disney	family	space	car
best	boring	laugh	house	voice	father	earth	jones
beautiful	worst	fun	scare	animated	son	sci-fi	fast
perfect	stupid	great	creepy	toy	mother	science	bad
excellent	money	hilarious	suspense	pixar	daughter	planet	agent
far	suppose	joke	ghost	adult	kevin	fiction	fun
enjoy	long	star	gore	child	boy	ship	diesel
long	awful	love	night	fun	young	ape	rock
wonderful	pretty	brooks	remake	princess	town	crew	chase

SLDA							
T1 (1.01)	T2 (-1.59)	T3 (0.73)	T4 (0.89)	T5 (0.91)	T6 (0.80)	T7 (0.70)	T8 (1.20)
enjoy	bad	funny	horror	voice	family	alien	classic
surprise	worst	comedy	scary	animation	father	space	era
entertaining	terrible	laugh	scare	disney	mother	earth	noir
fun	waste	joke	house	animated	son	science	today
nice	horrible	hilarious	creepy	toy	child	planet	early
pretty	awful	fun	atmosphere	child	daughter	fiction	silent
great	worse	parody	ghost	adult	young	sci-fi	modern
enjoyable	stupid	satire	haunt	pixar	parent	ship	kane
interesting	boring	gag	gore	cartoon	boy	predator	simple
definitely	crap	silly	disturbing	age	brother	scientist	citizen

HFT							
T1	T2	T3	T4	T5	T6	T7	T8
action	vampire	comedy	horror	animation	father	sci-fi	hulk
franchise	dracula	funny	halloween	disney	son	science	fox
installment	twilight	sandler	slasher	pixar	fanning	space	banner
diesel	blade	hilarious	scary	animated	dakota	mars	car
explosion	beckinsale	ferrell	eli	wall-e	dad	spaceship	bana
sequel	helsing	laugh	scare	costner	precious	planet	ross
cgi	underworld	wedding	house	nemo	boy	earth	wax
stunt	jacob	joke	myers	chicken	mother	robot	racing
vin	bella	gag	creepy	toy	bike	scientist	eric
fun	edward	comedic	carrie	dreamworks	parent	alien	norton

Table 3: 8 topics extracted with LDA-R-C, SLDA and HFT, $K = 100$ and the associated score for SLDA (see Mcauliffe and Blei [2008] for details).

Both LDA-R-C and SLDA extract qualitative topics, while we could not extract qualitative topics with HFT. The descriptive topics of the three methods are consistent around genres, sequels, actors or directors. We observe that the two methods LDA-R-C and SLDA extract similar topics (Table 3, topics T1 to T5 extracted with LDA-R-C are similar to topics T3 to T7 extracted with SLDA). From topics extracted with LDA-R-C, T1 is consistent around *comedy*, T2 around *horror*, T3 around *animation*, T4 around *family*, T5 around *science-fiction*. We can see the same consistency for topics T3 to T7 with SLDA. We observe that topics extracted with LDA-R-C share more top words with SLDA than with HFT. Indeed, the inference schemes of LDA-R-C and SLDA are close (i.e., variational EM), leading to similar topics. For instance, in Table 3, 6 out of 10 top words of topic T1 obtained with LDA-R-C also appear in the top of SLDA's topic T3. In the same way,

topics T2 to T5 extracted with LDA-R-C are respectively closer to topics T4 to T7 extracted with SLDA than topics T4 to T7 extracted with HFT.

The difference between LDA-R-C and SLDA is that with LDA-R-C we obtain two distinct sets of words. A first set of words that illustrates the rating of the users—corresponding to topics Q1 and Q2 in Table 3—and a second set of words that the user would employ to describe a movie—topics T1 to T6 in Table 3. There is no such distinction in SLDA as the output is a single list of topics associated with a score. We notice that most of the extracted topics with SLDA are descriptive—e.g., topics T3 to T8 in Table 3—and qualitative topics are associated to high scores (in absolute value)—e.g., topics T1 and T2 in Table 3. However, some descriptive topics are associated with high scores—e.g., topic T8 in Table 3—which makes difficult the distinction between qualitative and descriptive topics. As our model LDA-R-C extracts two sets of topics, there is no possible confusion between descriptive and qualitative topics. SLDA extracts a single list of topics and it is difficult nay impossible to automatically classify the topics as a qualitative or a descritpive.

In HFT, the parameters of LDA are linked to rating prediction parameters. As a result, the top words of the topics are still centered around generic genres, sequels, actors, directors but also contain words related to specific movies. For instance, in Table 3, the topic T4 extracted with HFT is centered around *comedy* and contains the words *sandler*, *ferrell* which are specific actor names and *wedding* which is a specific part of a plot. In the topic T5 extracted with HFT, centered around *animation movies*, we find the words *wall-e*, *nemo* which are specific titles and *costner* which is an actor name. The top words in both LDA-R-C and SLDA topics are more generic, leading to better predictions. Indeed, it is more likely that a review about a *comedy* movie contains *funny* than *wedding*, as only few comedy movies are related to a wedding.

2.4 RELATED WORK

Several techniques have been proposed to extract information from raw text data. LDA [Blei et al., 2003] is a probabilistic model that infers hidden topics given a text corpus where each document of the corpus can then be represented as topic probabilities. The assumption behind LDA is that each document is generated from a mixture of topics and the model infers the hidden topics and the topic proportions of each document. For increasing consistency of inferred topics, a regularized version of LDA is proposed by Newman et al. [2011]. This regularized version puts structured priors on the hidden topics. The parameters of the prior are pre-computed and consist in a "covariance" matrix which captures the short-range dependencies between words. This matrix has a regularization effect on the topics. The authors compare different priors. We increase the consistency of topics even further by aggregating reviews in LDA. Titov and McDonald [2008] present a LDA based model with two types of topics; this model infers global topics that contain the different types of movie being reviewed, while the

local topics extract the specific aspects of the movies. Instead, we chose to add qualitative information in the LDA topics, similar to the model proposed by Lin and He [2009] where a sentiment label is inferred for each document. The difference with our method is that we leverage the ratings found in movie reviews datasets to extract qualitative and descriptive information in separate topics.

In the rating prediction area, McAuley and Leskovec [2013] propose a transformation between LDA parameters and collaborative filtering parameters. Numerical ratings and words are processed in two separated models. Instead, we combine scores and words in the same LDA model. The model proposed by Mcauliffe and Blei [2008] infers parameters from both a text corpus and numerical ratings. The model is then able to predict a score given a new text. We use a similar approach to predict directly a list of words instead of numerical ratings. The model proposed by Ling et al. [2014] combines LDA with matrix factorization to predict ratings. The main difference with our approach is that we distinguish qualitative and descriptive words in the topics while the topics inferred by Ling et al. [2014] mix qualitative and descriptive words. Their model is also suited for rating prediction while we focus on word prediction.

2.5 CONCLUSION

We have proposed six LDA-based models for word prediction from crowdsourced reviews and ratings. We show on an IMDB dataset that our `LDA-R-C` model combining movie profiling and ratings performs slightly better than the state of the art. It builds a set of descriptive topics that convey the features of movies—e.g., genres, actors, directors—and contain the words the user would employ to describe a movie. It also builds a set of qualitative topics that convey the opinion of users about movies and contain the words that influence—positively or negatively—the final ratings of users.

While studying our `LDA-U` model, we came to the conclusion that it is difficult to build a user profile as each user writes reviews about very different movies expressing very different opinions.

For now, our models only extract two qualitative topics (positive and negative). We plan to build models that would extract a wider range of qualitative topics by reducing the observed ratings to a wider range of values than $\{+1, -1\}$.

The review prediction is currently based on single words prediction, which is not intuitive for users. We plan to predict readable sentences which would facilitate evaluation by users with A/B testing. Implemented with readable reviews or tags, our model could be integrated in a recommender to provide a personalized opinion summary to users for each content.

APPENDIX

2.A VARIATIONAL DERIVATION OF LDA-R-C

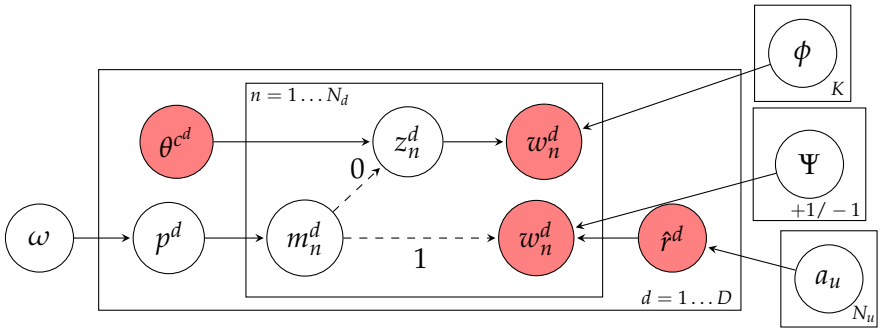


Figure 5: Graphical representation of the model LDA-R-C including reduced ratings applied on D documents. White nodes represent hidden variables and colored nodes represent observed variables. The observed rating r^d is not reported for the sake of clarity.

In this section, we provide the full variational derivation of our model LDA-R-C presented Figure 5. Our objective is to maximize the likelihood of the observed corpus of documents $\mathcal{W} = \{\mathbf{w}^1, \dots, \mathbf{w}^D\}$:

$$p(\mathcal{W}|\omega, \{\theta^{c^d}\}_d, \eta, \{\hat{r^d}\}_d) = \prod_{d=1}^{D} p(\mathbf{w}^d|\omega, \theta^{c^d}, \eta, \hat{r}^d).$$

As this likelihood is intractable to compute, we maximize an approximation of the likelihood $\mathcal{L}(q) = \sum_{d=1}^{D} \mathcal{L}^d(q)$ over a variational family of distributions. Following Hoffman et al. [2013], we have for any $\mathbf{w}^d \in \mathcal{W}$:

$$\begin{aligned}\log p(\mathbf{w}^d|\omega, \theta^{c^d}, \eta, \hat{r}^d) \geq & \mathbb{E}_q[\log p(\mathbf{w}^d, z^d, m^d, p^d, \phi, \Psi|\omega, \theta^{c^d}, \eta, \hat{r}^d)] \\ & - \mathbb{E}_q[\log q(z^d, m^d, p^d, \phi, \Psi)] \\ \equiv & \mathcal{L}^d(q),\end{aligned}$$

where q represents the variational model. We choose the variational model q to be in the meanfield variational family:

$$q(z^d, m^d, p^d, \phi, \Psi) = q(\phi|\lambda)q(\Psi|\Lambda)q(p^d|\pi^d)\prod_{n=1}^{N} q(z_n^d|\alpha_n^d)q(m_n^d|\mu_n^d),$$

with, $\forall d = 1, \dots, D$:

- $q(\phi^k|\lambda^k) \sim$Dirichlet(λ^k) with $\lambda^k \in \mathbb{R}^V$ and $k = 1, \dots, K$,

- $q(\Psi^s|\Lambda^s) \sim$Dirichlet(Λ^s) with $\Lambda^s \in \mathbb{R}^V$ and $s \in \{+1, -1\}$,
- $q(p^d|\pi^d) \sim$Dirichlet(π^d) with $\pi^d \in \mathbb{R}^2$,
- $q(z_n^d|\alpha_n^d) \sim$Multinomial(α_n^d) with $\alpha_n^d \in \mathbb{R}^K$, $\sum_k \alpha_{n,k}^d = 1$ and $n = 1, \ldots, N$,
- $q(m_n^d|\mu_n^d) \sim$Multinomial(μ_n^d) with $\mu_n^d \in \mathbb{R}^2$, $\mu_{n,1}^d + \mu_{n,2}^d = 1$ and $n = 1, \ldots, N$.

We also have:

$$p(\mathbf{w}^d, z^d, m^d, p^d, \phi, \Psi|\omega, \theta^{c^d}, \eta, \hat{r}^d) = p(\phi|\eta)p(\Psi|\eta)p(p^d|\omega) \times \prod_{n=1}^{N} p(w_n^d|\phi, \Psi, z_n^d, r^d, m_n^d)p(z_n^d|\theta^{c^d})p(m_n^d|p^d),$$

with, $\forall d = 1, \ldots, D$:

- $p(\phi^k|\eta) \sim$Dirichlet($\eta\mathbf{1}$) with $\eta \in \mathbb{R}$ and $k = 1, \ldots, K$,
- $p(\Psi^s|\eta) \sim$Dirichlet($\eta\mathbf{1}$) with $\eta \in \mathbb{R}$ and $s \in \{+1, -1\}$,
- $p(p^d|\omega) \sim$Dirichlet(ω) with $\omega \in \mathbb{R}^2$,
- $p(z_n^d|\theta^{c^d}) \sim$Multinomial($\theta^{c^d}$) with $\theta^{c^d} \in \mathbb{R}^K$, $\sum_k \theta_k^{c^d} = 1$ and $n = 1, \ldots, N$,
- $p(m_n^d|p^d) \sim$Multinomial(p^d) with $p^d \in \mathbb{R}^2$, $p_1^d + p_2^d = 1$ and $n = 1, \ldots, N$,
- $p(w_n^d|\phi, \Psi, z_n^d, r^d, m_n^d) \sim$Multinomial$\Big(\phi^{z_n^d}\mathbf{1}[m_n^d = 0] + \Psi^{r^d}\mathbf{1}[m_n^d = 1]\Big)$.

We then maximize $\mathcal{L}(q)$ by iteratively maximizing $\mathcal{L}(q)$ with respect to variational parameters $\lambda, \Lambda, \pi, \alpha, \mu$ (E-step) then maximizing $\mathcal{L}(q)$ with respect to hyperparameters ω, η (M-step).

2.A.1 Variational E-step

For the E-step, we maximize $\mathcal{L}(q)$ with respect to variational parameters λ, Λ, π, α, μ by alternatively setting the gradient of $\mathcal{L}(q)$ with respect to each paramater to zero. It gives the following updates for the variational parameters, for $n = 1, \ldots, N$, $k = 1, \ldots, K$, $i = 1, 2$ and $s \in \{-1, +1\}$:

$$\begin{cases} \alpha_{n,k}^d & \propto & \theta_k^{c^d} \exp\left[\mu_{n,1}^d \left(\psi(\lambda_{w_n^d}^k) - \psi(\sum_j \lambda_j^k)\right)\right], \\ \pi_i^d & = & \omega_i + \sum_{n=1}^N \mu_{n,i}^d, \\ \mu_{n,1}^d & \propto & \exp\left[\psi(\pi_1^d) + \sum_{k=1}^K \psi(\lambda_{w_n^d}^k) - \psi(\sum_j \lambda_j^k)\right], \\ \mu_{n,2}^d & \propto & \exp\left[\psi(\pi_2^d) + \sum_{s\in\{-1,+1\}} \psi(\Lambda_{w_n^d}^s) - \psi(\sum_j \Lambda_j^s)\right], \\ \\ \lambda_v^k & = & \eta + \sum_{d=1}^D \sum_{n=1}^{N_d} \mu_{n,1}^d \alpha_{n,k}^d \mathbf{1}[w_n^d = v], \\ \Lambda_v^s & = & \eta + \sum_{d:r^d=s} \sum_{n=1}^{N_d} \mu_{n,2}^d \mathbf{1}[w_n^d = v]. \end{cases}$$

ψ is the digamma function: $\psi(x) = \frac{d}{dx} \ln \Gamma(x)$.

2.A.2 Variational M-step

For the M-step, we maximize $\mathcal{L}(q)$ with respect to the hyperparameters ω, η. We use the Newton method for each parameter, using the same scheme than in LDA Blei et al. [2003]. We have the following derivatives for ω:

$$\begin{cases} \frac{\partial}{\partial \omega_i} \mathcal{L}(q) & = & D \left(\psi(\sum_j \omega_j) - \psi(\omega_i) \right) + \sum_{d=1}^{D} \left(\psi(\pi_i^d), -\psi(\sum_j \pi_j^d) \right) \\ \frac{\partial^2}{\partial \omega_i \partial \omega_j} \mathcal{L}(q) & = & D\psi'(\sum_l \omega_l) - \mathbf{1}[i=j] D\psi'(\omega_i). \end{cases}$$

We have the following derivatives for η

$$\begin{cases} \frac{\partial}{\partial \eta} \mathcal{L}(q) & = & (K+2)V \left(\psi(V\eta) - \psi(\eta) \right) + \sum_{v=1}^{V} \left(\sum_{k=1}^{K} \psi(\lambda_v^k) + \sum_{s \in \{-1,+1\}} \psi(\Lambda_v^s) \right) \\ & & -V \left(\sum_{k=1}^{K} \psi(\sum_{v=1}^{V} \lambda_v^k) + \sum_{s \in \{-1,+1\}} \psi(\sum_{v=1}^{V} \Lambda_v^s) \right), \\ \frac{\partial^2}{\partial \eta^2} \mathcal{L}(q) & = & (K+2)V \left(V\psi'(V\eta) - \psi'(\eta) \right). \end{cases}$$

We maximize $\mathcal{L}(q)$ with respect to ω by doing iterations of Newton steps until convergence:

$$\omega^{(t+1)} = \omega^{(t)} - H^{-1} \nabla_{\omega^{(t)}} \mathcal{L}(q),$$

where H is the Hessian $H = \nabla^2_{\omega^{(t)}} \mathcal{L}(q)$. We then maximize $\mathcal{L}(q)$ with respect to η by again doing iterations of Newton steps until convergence:

$$\eta^{(t+1)} = \eta^{(t)} - \left[\frac{\partial^2}{(\partial \eta^{(t)})^2} \mathcal{L}(q) \right]^{-1} \left(\frac{\partial}{\partial \eta^{(t)}} \mathcal{L}(q) \right).$$

3

ONLINE BUT ACCURATE INFERENCE FOR LATENT VARIABLE MODELS WITH LOCAL GIBBS SAMPLING

Probabilistic graphical models provide general modelling tools for complex data, where it is natural to include assumptions on the data generating process by adding latent variables in the model. Such latent variable models are adapted to a wide variety of unsupervised learning tasks [Koller and Friedman, 2009, Murphy, 2012]. In this chapter, we focus on parameter inference in such latent variable models where the main operation needed for the standard expectation-maximization (EM) algorithm is intractable, namely dealing with conditional distributions over latent variables given the observed variables; latent Dirichlet allocation (LDA) [Blei et al., 2003] is our motivating example, but many hierarchical models exhibit this behavior, e.g., ICA with heavy-tailed priors. For such models, there exist two main classes of methods to deal efficiently with intractable exact inference in large-scale situations: sampling methods or variational methods.

Sampling methods can handle arbitrary distributions and lead to simple inference algorithms while converging to exact inference. However it may be slow to converge and non scalable to big datasets in practice. In particular, although efficient implementations have been developed, for example for LDA [Zhao et al., 2014, Yan et al., 2009], MCMC methods may not deal efficiently yet with continuous streams of data for our general class of models.

On the other hand, *variational inference* builds an approximate model for the posterior distribution over latent variables—called variational—and infer parameters of the true model through this approximation. The fitting of this variational distribution is formulated as an optimization problem where efficient (deterministic) iterative techniques such as gradient or coordinate ascent methods apply. This approach leads to scalable inference schemes [Hoffman et al., 2013], but due to approximations, there always remains a gap between the variational posterior and the true posterior distribution, inherent to algorithm design, and that will not vanish when the number of samples and the number of iterations increase.

Beyond the choice of approximate inference techniques for latent variables, parameter inference may be treated either from the *frequentist* point of view, e.g., using maximum likelihood inference, or a *Bayesian* point of view, where the posterior distribution of the parameter given the observed data is approximated. With massive numbers of observations, this posterior distribution is typically peaked around the maximum likelihood estimate, and the two inference frameworks should not differ much [Van der Vaart, 2000].

In this chapter, we focus on methods that make a single pass over the data to estimate parameters. We make the following contributions:

1. We review and compare existing methods for online inference for latent variable models from a non-canonical exponential family in Section 3.1, and draw explicit links between several previously proposed frequentist or Bayesian methods. Given the large number of existing methods, our unifying framework allows one to understand differences and similarities between all of them.

2. We propose in Section 3.2 a novel inference method for the frequentist estimation of parameters, that adapts MCMC methods to online inference of latent variable models with the proper use of "local" Gibbs sampling. In our online scheme, we apply Gibbs sampling to the current observation, which is "local", as opposed to "global" batch schemes where Gibbs sampling is applied to the entire dataset.

3. After formulating LDA as a non-canonical exponential family in Section 3.3, we provide an extensive set of experiments in Section 3.5, where our new approach outperforms all previously proposed methods. In particular, using Gibbs sampling for latent variable inference is superior to variational inference in terms of test log-likelihoods. Moreover, Bayesian inference through variational methods perform poorly, sometimes leading to worse fits with latent variables of higher dimensionality.

This work is under revision at JMLR.

3.1 ONLINE EM

We consider an *exponential family* model on random variables (X, h) with parameter $\eta \in \mathcal{E} \subseteq \mathbb{R}^d$ and with density [Lehmann and Casella, 1998]:

$$p(X, h|\eta) = a(X, h) \exp\left[\langle \phi(\eta), S(X, h)\rangle - \psi(\eta)\right]. \tag{1}$$

We assume that h is hidden and X is observed. The vector $\phi(\eta) \in \mathbb{R}^d$ represents the natural parameter, $S(X, h) \in \mathbb{R}^d$ is the vector of sufficient statistics, $\psi(\eta)$ is the log-normalizer, and $a(X, h)$ is the underlying base measure. We consider a *non-canonical* family as in many models (such as LDA), the natural parameter $\phi(\eta)$ does not coincide with the model parameter η, that is, $\phi(\eta) \not\equiv \eta$; we however assume that ϕ is injective.

We consider N *i.i.d.* observations $(X_i)_{i=1,\dots,N}$ from a distribution $t(X)$, which may be of the form $P(X|\eta^*) = \int_h p(X, h|\eta^*)\mathrm{d}h$ for our model above and a certain $\eta^* \in \mathcal{E}$ (well-specified model) or not (misspecified model). Our goal is to obtain a predictive density $r(X)$ built from the data and using the model defined in (1), with the maximal expected log-likelihood $\mathbb{E}_{t(X)} \log r(X)$.

3.1.1 Maximum likelihood estimation

In the frequentist perpective, the predictive distribution $r(X)$ is of the form $p(X|\hat{\eta})$, for a well-defined estimator $\hat{\eta} \in \mathcal{E}$. The most common method is the EM algorithm [Dempster et al., 1977], which is an algorithm that aims at maximizing the likelihood of the observed data, that is,

$$\max_{\eta \in \mathcal{E}} \sum_{i=1}^{N} \log p(X_i|\eta). \tag{2}$$

More precisely, the EM algorithm is an iterative process to find the maximum likelihood (ML) estimate given observations $(X_i)_{i=1,\dots,N}$ associated to hidden variables $(h_i)_{i=1,\dots,N}$. It may be seen as the iterative construction of lower bounds of the log-likelihood function [Bishop, 2006]. In the exponential family setting (1), we have, by Jensen's inequality, given the model defined by $\eta' \in \mathcal{E}$ from the previous iteration, and for any parameter $\eta \in \mathcal{E}$:

$$\begin{aligned}
\log p(X_i|\eta) &= \log \int p(X_i, h_i|\eta) dh_i \\
&\geq \int p(h_i|X_i, \eta') \log \frac{p(X_i, h_i|\eta)}{p(h_i|X_i, \eta')} dh_i \\
&= \int p(h_i|X_i, \eta') \left(\langle \phi(\eta), S(X_i, h_i) \rangle - \psi(\eta) \right) dh_i - C_i(\eta') \\
&= \langle \phi(\eta), \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i, h_i) \right] \rangle - \psi(\eta) - C_i(\eta'),
\end{aligned}$$

for a certain constant $C_i(\eta')$, with equality if $\eta' = \eta$. Thus, EM-type algorithms build locally tight lower bounds of the log-likelihood in (2), which are equal to

$$\langle \phi(\eta), \textstyle\sum_{i=1}^{N} s_i \rangle - N\psi(\eta) + \text{cst},$$

for appropriate values of $s_i \in \mathbb{R}^d$ obtained by computing conditional expectations with the distribution of h_i given X_i for the current model defined by η' (E-step), i.e., $s_i = \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i, h_i) \right]$. Then this function of η is maximized to obtain the next iterate (M-step). In standard EM applications, these two steps are assumed tractable. In Section 3.2, we will only assume that the M-step is tractable while the E-step is intractable.

Standard EM will consider $s_i = \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i, h) \right]$ for the previous value of the parameter η for all i, and hence, at every iteration, all observations X_i, $i = 1, \dots, N$ are considered for latent variable inference, leading to a slow "batch" algorithm for large N.

Incremental EM [Neal and Hinton, 1998] will only update a single element s_i coming from a single observation X_i and update the corresponding part of the sum $\sum_{j=1}^{N} s_j$ without changing other elements. In the extreme case where a single pass over the data is made, then the M-step at iteration i maximizes

$$\langle \phi(\eta), \sum_{j=1}^{i} \mathbb{E}_{p(h_j|X_j,\eta_{j-1})} \left[S(X_j, h_j) \right] \rangle - i\psi(\eta),$$

with respect to η. In the next section, we provide a (known) other interpretation of this algorithm.

3.1.2 Stochastic approximation

Given our frequentist objective $\mathbb{E}_{t(X)} \log p(X|\eta)$ to maximize defined as an expectation, we may consider two forms of stochastic approximation [Kushner and Yin, 2003], where observations X_i sampled from $t(X)$ are processed only once. The first one is stochastic gradient ascent, of the form

$$\eta_i = \eta_{i-1} + \rho_i \frac{\partial \log p(X_i|\eta)}{\partial \eta},$$

or appropriately renormalized version thereof, i.e., $\eta_i = \eta_{i-1} + \rho_i H^{-1} \frac{\partial \log p(X_i|\eta)}{\partial \eta}$, with several possibilities for the $d \times d$ matrix H, such as the negative Hessian of the partial or the full log-likelihood, or the negative covariance matrix of gradients, which can be seen as versions of natural gradient—see Titterington [1984], Delyon et al. [1999], Cappé and Moulines [2009]. This either leads to slow convergence (without H) or expensive iterations (with H), with the added difficulty of choosing a proper scale and decay for the step-size ρ_i.

A key insight of Delyon et al. [1999], Cappé and Moulines [2009] is to use a different formulation of stochastic approximation, *not explicitly based on stochastic gradient ascent*. They consider the stationary equation $\mathbb{E}_{t(X)} \left[\frac{\partial \log p(X|\eta)}{\partial \eta} \right] = 0$ and expand it using the exponential family model (1) as follows:

$$\begin{aligned} \frac{\partial \log p(X|\eta)}{\partial \eta} &= \frac{\partial \log \int p(X,h|\eta) dh}{\partial \eta} \\ &= \phi'(\eta) \mathbb{E}_{p(h|X,\eta)} \left[S(X,h) \right] - \psi'(\eta). \end{aligned}$$

Given standard properties of the exponential family, namely

$$\psi'(\eta) = \phi'(\eta) \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right] [1],$$

and assuming invertibility of $\phi'(\eta)$, this leads to the following stationary equation:

$$\mathbb{E}_{t(X)} \left[\mathbb{E}_{p(h|X,\eta)} \left[S(X,h) \right] \right] = \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right].$$

[1] Proof: Given (1), $\int_{X,h} p(X,h|\eta) \mathrm{d}(X,h) = 1 \Rightarrow \psi(\eta) = \log \left[\int_{X,h} a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h) \right]$. We then derive this identity with respect to η, which gives:

$$\begin{aligned} \psi'(\eta) &= \frac{\int_{X,h} \phi'(\eta) S(X,h) a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)}{\int_{X,h} a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)} \\ &= \frac{\phi'(\eta) \int_{X,h} S(X,h) a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)}{e^{\psi(\eta)}} \\ &= \phi'(\eta) \int_{X,h} S(X,h) p(X,h|\eta) \mathrm{d}(X,h) \\ &= \phi'(\eta) \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right]. \end{aligned}$$

This stationary equation states that at optimality the sufficient statitics have the same expectation for the full model $p(h, X|\eta)$ and the joint "model/data" distribution $t(X)p(h|X, \eta)$.

Another important insight of Delyon et al. [1999], Cappé and Moulines [2009] is to consider the change of variable $s(\eta) = \mathbb{E}_{p(h,X|\eta)}[S(X, h)]$ on sufficient statistics, which is equivalent to

$$\eta = \eta^*(s) \in \arg\max \langle \phi(\eta), s\rangle - \psi(\eta),$$

(which is the usual M-step update). See Cappé and Moulines [2009] for detailed assumptions allowing this inversion. We may then rewrite the equation above as

$$\mathbb{E}_{t(X)}\big(\mathbb{E}_{p(h|X,\eta^*(s))}[S(X, h)]\big) = s.$$

This is a non-linear equation in $s \in \mathbb{R}^d$, with an expectation with respect to $t(X)$ which is only accessed through i.i.d. samples X_i, and thus a good candidate for the Robbins-Monro algorithm to solve stationary equations (and not to minimize functions) [Kushner and Yin, 2003], which takes the simple form:

$$s_i = s_{i-1} - \rho_i\big(s_{i-1} - \mathbb{E}_{p(h_i|X_i,\eta^*(s_{i-1}))}[S(X_i, h_i)]\big),$$

with a step-size ρ_i. It may be rewritten as

$$\left\{\begin{array}{l} s_i = (1-\rho_i)s_{i-1} + \rho_i \mathbb{E}_{p(h_i|X_i,\eta_{i-1})}[S(X_i, h_i)] \\ \eta_i = \eta^*(s_i), \end{array}\right. \tag{3}$$

which has a particularly simple interpretation: instead of computing the expectation for all observations as in full EM, this stochastic version keeps tracks of old sufficient statistics through the variable s_{i-1} which is updated towards the current value $\mathbb{E}_{p(h_i|X_i,\eta_{i-1})}[S(X_i, h_i)]$. The parameter η is then updated to the value $\eta^*(s_i)$. Cappé and Moulines [2009] show that this update is asymptotically equivalent to the natural gradient update with three main improvements: (a) no matrix inversion is needed, (b) the algorithm may be accelerated through Polyak-Ruppert averaging [Polyak and Juditsky, 1992], i.e., using the average $\bar{\eta}_N$ of all η_i instead of the last iterate η_N, and (c) the step-size is particularly simple to set, as we are taking *convex* combinations of sufficient statistics, and hence only the decay rate of ρ_i has to be chosen, i.e., of the form $\rho_i = i^{-\kappa}$, for $\kappa \in (0, 1]$, without any multiplicative constant.

INCREMENTAL VIEW. For the specific stepsize $\rho_i = 1/i$, the online EM algorithm (3) corresponds exactly to the incremental EM presented above [Neal and Hinton, 1998], as then

$$s_i = \frac{1}{i}\sum_{j=1}^{i} \mathbb{E}_{p(h_j|X_j,\eta_{j-1})}[S(X_j, h_j)].$$

See Mairal [2014] for a detailed convergence analysis of incremental algorithms, in particular showing that step-sizes larger than $1/i$ are preferable (we observe this in practice in Section 3.5).

MONTE CARLO METHODS. There exist alternative methods to the EM algorithm based on Monte Carlo sampling to compute the maximum likelihood. For instance, the Monte Carlo EM method (MCEM) [Wei and Tanner, 1990] is a general Bayesian approach (i.e., η is a random variable) to approximate the maximizer of the posterior distribution $p(\eta|X,h)$ with Monte Carlo sampling. More precisely, in the MCEM method, similarly to EM, a surrogate function of the log-likelihood is used, given by:

$$Q(\eta,\eta_t) = \int_h \log[p(\eta|X,h)]p(h|X,\eta_t)\mathrm{d}h.$$

The function Q is approximated by sampling the latent variables h from the current conditional $p(h|X,\eta_t)$:

$$\hat{Q}(\eta,\eta_t) = \sum_{i=1}^{P} \log p(\eta|X,h^i),$$

where $(h^i)_{i=1,\dots,P}$ are the samples drawn from the conditional $p(h|X,\eta_t)$. The approximation $\hat{Q}$ is then maximized with respect to η. Note that this method is a bacth method, namely, samples are drawn over all the dataset.

Other sequential Monte Carlo methods (SMC) apply particle methods to the approximation of the posterior $p(\eta|X,h)$ [Kantas et al., 2015].

The two Monte Carlo methods mentionned above also consist in sufficient statistics updates for the class of models considered here.

3.2 ONLINE EM WITH INTRACTABLE MODELS

The online EM updates in (3) lead to a scalable algorithm for optimization when the local E-step is tractable. However, in many latent variable models—e.g., LDA, hierarchical Dirichlet processes [Teh et al., 2006], or ICA [Hyvärinen et al., 2004]—it is intractable to compute the conditional expectation $\mathbb{E}_{p(h|X,\eta)}[S(X,h)]$.

Following Rohde and Cappé [2011], we propose to leverage the scalability of online EM updates (3) and locally approximate the conditional distribution $p(h|X,\eta)$ in the case this distribution is intractable to compute. We will however consider different approximate methods, namely Gibbs sampling or variational inference. Our method is thus restricted to models where the hidden variable h may naturally be splitted in two or more groups of simple random variables. Our algorithm is described in Algorithm 1 and may be instantiated with two approximate inference schemes which we now describe.

Algorithm 1 Gibbs / Variational online EM

Input: η_0, s_0, $\kappa \in (0, 1]$.
for $i = 1, \dots, N$ **do**
- Collect observation X_i,
- Estimate $p(h_i|X_i, \eta_{i-1})$ with sampling (`G-OEM`) or variational inference (`V-OEM`),
- Apply (3) to sufficient statistics s_i and parameter η_i with $\rho_i = 1/i^\kappa$,

end for
Output: $\bar{\eta}_N = \frac{1}{N}\sum_{i=1}^N \eta_i$ or η_N.

3.2.1 Variational inference: `V-OEM`

While variational inference had been considered before for online estimation of latent variable models, in particular for LDA for incremental EM [Sato et al., 2010], using it for online EM (which is empirically faster) had not been proposed and allows us to use bigger step-sizes (e.g., $\kappa = 1/2$). These methods are based on maximizing the negative variational "free-energy"

$$\mathbb{E}_{q(h|\eta)}\left[\log \frac{p(X, h|\eta)}{q(h|\eta)}\right], \tag{4}$$

with respect to $q(h|\eta)$ having a certain factorized form adapted to the model at hand, so that efficient coordinate ascent may be used. See, e.g., Hoffman et al. [2013]. We now denote online EM with variational approximation of the conditional distribution $p(h|X, \eta)$ as `V-OEM`.

3.2.2 Sampling methods: `G-OEM`

MCMC methods to approximate the conditional distribution of latent variables with online EM have been considered by Rohde and Cappé [2011], who apply locally the Metropolis-Hasting (M-H) algorithm [Metropolis et al., 1953, Hastings, 1970], and show results on simple synthetic datasets. While Gibbs sampling is widely used for many models such as LDA due to its simplicity and lack of external parameters, M-H requires a proper proposal distribution with frequent acceptance and fast mixing, which may be hard to find in high dimensions. We provide a different simpler local scheme based on Gibbs sampling (thus adapted to a wide variety of models), and propose a thorough favorable comparison on synthetic and real datasets with existing methods.

The Gibbs sampler is used to estimate posterior distributions by alternatively sampling parts of the variables given the other ones [see Casella and George, 1992, for details], and is standard and easy to use in many common latent variable models. In the following, the online EM method with Gibbs estimation of the conditional distribution $p(h|X, \eta)$ is denoted `G-OEM`.

As mentioned above, the online EM updates correspond to a stochastic approximation algorithm and thus are robust to random noise in the local E-step. As a result, our sampling method is particularly adapted as it is a random estimate of the E-step—see a theoretical analysis by Rohde and Cappé [2011], and thus we only need to compute a few Gibbs samples for the estimation of $p(h|X_i, \eta_{i-1})$. A key contribution of our work is to reuse sampling techniques that have proved competitive in the batch set-up and to compare them to existing variational approaches.

3.2.3 "Boosted" inference

As the variational and MCMC estimations of $p(h|X_i, \eta_{i-1})$ are done with iterative methods, we can boost the inference of Algorithm 1 by applying the update in the parameter η in (3) after each iteration of the estimation of $p(h|X_i, \eta_{i-1})$. In the context of LDA, this was proposed by Sato et al. [2010] for incremental EM and we extend it to all versions of online EM. With this boost, we expect that the global parameters η converge faster, as they are updated more often. In the following, we denote by `G-OEM++` (resp. `V-OEM++`) the method `G-OEM` (resp. `V-OEM`) augmented with this boost.

3.2.4 Variational Bayesian estimation

In the Bayesian perspective where η is seen as a random variable, we either consider a distribution based on model averaging, e.g., $r(X) = \int p(X|\eta)q(\eta)d\eta$ where $q(\eta) \propto \prod_{i=1}^{N} p(X_i|\eta)p(\eta)$ is the posterior distribution, or

$$r(X) = p(X|\bar{\eta}),$$

where $\bar{\eta}$ is the summary (e.g., the mean) of the posterior distribution $q(\eta)$, or of an approximation, which is usually done in practice [see, e.g., Hoffman and Blei, 2015] and is asymptotically equivalent when N tends to infinity.

The main problem is that, *even when the conditional distribution of latent variables is tractable*, it is intractable to manipulate the *joint* posterior distribution over the latent variables $h_1, \ldots, h_N$, and the parameter η. Variational inference techniques consider an approximation where hidden variables are independent of the parameter η, i.e., such that

$$p(\eta, h_1, \ldots, h_N|X_1, \ldots, X_N) \approx q(\eta) \prod_{i=1}^{N} q(h_i),$$

which corresponds to the maximization of the following lower bound—called Evidence Lower BOund (ELBO)—on the log-likelihood $\log p(X_1, \ldots, X_n)$ [Bishop, 2006]:

$$\int q(\eta) \prod_{i=1}^{N} q(h_i) \log \frac{p(\eta) \prod_{i=1}^{n} p(X_i, h_i|\eta)}{q(\eta) \prod_{i=1}^{N} q(h_i)} d\eta dh_1 \cdots dh_N.$$

The key insight from Hoffman et al. [2010], Broderick et al. [2013] is to consider the variational distribution $q(\eta)$ as the global parameter, and the cost function above as a sum of *local* functions that depend on the data X_i and the variational distribution $q(h_i)$. Once the local variational distribution $q(h_i)$ is maximized out, the sum structure may be leveraged in similar ways than for frequentist estimation, either by direct (natural) stochastic gradient [Hoffman et al., 2010] or incremental techniques that accumulate sufficient statistics [Broderick et al., 2013]. A nice feature of these techniques is that they extend directly to models with intractable latent variable inference, by making additional assumptions on $q(h_i)$ (see for example the LDA situation in Section 3.3).

In terms of actual updates, they are similar to online EM in Section 3.2.1, with a few changes, but which turn out to lead to significant differences in practice. The similarity comes from the expansion of the ELBO as

$$\mathbb{E}_{q(\eta)}\left[\sum_{i=1}^{N}\mathbb{E}_{q(h_i)}\log\frac{p(X_i,h_i|\eta)}{q(h_i)}\right]+\mathbb{E}_{q(\eta)}\left[\log\frac{p(\eta)}{q(\eta)}\right].$$

The left hand side has the same structure than the variational EM update in (4), thus leading to similar updates, while the right hand side corresponds to the "Bayesian layer", and the maximization with respect to $q(\eta)$ is similar to the M-step of EM (where η is seen as a parameter).

Like online EM techniques presented in Section 3.2, approximate inference for latent variable is used, but, when using Bayesian stochastic variational inference techniques, there are two additional sources of inefficiencies: (a) extra assumptions regarding the independence of η and $h_1,\dots,h_N$, and (b) the lack of explicit formulation as the minimization of an expectation, which prevents the simple use of the most efficient stochastic approximation techniques (together with their guarantees). While (b) can simply slow down the algorithm, (a) may lead to results which are far away from exact inference, even for large numbers of samples (see examples in Section 3.5).

Beyond variational inference, Gibbs sampling has been recently considered by Gao et al. [2016]: their method consists in sampling hidden variables for the current document given current parameters, but (a) only some of the new parameters are updated by incrementally aggregating the samples of the current document with current parameters, and (b) the method is slower than `G-OEM` (see Section 3.5).

3.3 Application to LDA

LDA [Blei et al., 2003] is a probabilistic model that infers hidden topics given a text corpus where each document of the corpus can be represented as topic probabilities. In particular, the assumption behind LDA is that each document is generated from a mixture of topics and the model infers the hidden topics and the topic proportions of each document. In practice, inference is done using Bayesian variational EM [Blei et al., 2003], Gibbs sampling [Griffiths and Steyvers,

2004, Wallach, 2006] or stochastic variational inference [Hoffman et al., 2010, Broderick et al., 2013, Sato et al., 2010].

HIERARCHICAL PROBABILISTIC MODEL. Let $\mathcal{C} = \{X_1, \dots, X_D\}$ be a corpus of D documents, V the number of words in our vocabulary and K the number of latent topics in the corpus. Each topic β^k corresponds to a discrete distribution on the V words (that is an element of the simplex in V dimensions). A hidden discrete distribution θ_i over the K topics (that is an element of the simplex in K dimensions) is attached to each document X_i. As detailed in Chapter 1, LDA is a generative model applied to a corpus of text documents which assumes that each word of the i^{th} document X_i is generated as follows:

- Choose $\theta_i \sim \text{Dirichlet}(\alpha)$,
- For each word $x_n \in X_i = \left(x_1, \dots, x_{N_{X_i}}\right)$:
  - Choose a topic $z_n \sim \text{Multinomial}(\theta_i)$,
  - Choose a word $x_n \sim \text{Multinomial}(\beta^{z_n})$.

In our settings, an observation is a document $X_i = (x_1, \dots, x_{N_{X_i}})$ where for all $1 \leq n \leq N_{X_i}$, $x_n \in \{0,1\}^V$ and $\sum_{v=1}^V x_{nv} = 1$. Each observation X_i is associated with the hidden variables h_i, with $h_i \equiv (Z_i = (z_1, \dots, z_{N_{X_i}}), \theta_i)$. The vector θ_i represents the topic proportions of document X_i and Z_i is the vector of topic assignments of each word of X_i. The variable h_i is local, i.e., attached to one observation X_i. The parameters of the model are global, represented by $\eta \equiv (\beta, \alpha)$, where β represents the topic matrix and α represents the Dirichlet prior on topic proportions.

We derive the LDA model in Section 3.3.1 to find ϕ, S, ψ and a such that the joint probability $p(Z, \theta | X, \alpha, \beta)$ is in a non-canonical exponential family (1).

We may then readily apply all algorithms from Section 3.2 by estimating the conditional expectation $\mathbb{E}_{Z,\theta|X,\alpha,\beta}[S(X, Z, \theta)]$ with either variational inference (`V-OEM`) or Gibbs sampling (`G-OEM`). See Sections 3.3.2 and 3.3.3 for online EM derivations. Note that the key difficulty of LDA is the presence of two interacting hidden variables Z and θ.

3.3.1 LDA and exponential families

An observation X is a document of length N_X, where $X = (x_1, \dots, x_{N_X})$, each word is represented by $x_n \in \{0,1\}^V$ with $\sum_{v=1}^V x_{nv} = 1$. Our corpus $\mathcal{C}$ is a set of D observations $\mathcal{C} = (X_1, \dots, X_D)$. For each document X_i a hidden variable θ^i is associated, corresponding to the topic distribution of document X_i. For each word x_n of document X_i a hidden variable $z_n \in \{0,1\}^K$ is attached, corresponding to the topic assignment of word x_n. We want to find ϕ, S, ψ and a such that, the joint probability is in the exponential family (1):

$$p(X, Z, \theta | \beta, \alpha) = a(X, Z, \theta) \exp\left[\langle \phi(\beta, \alpha), S(X, Z, \theta) \rangle - \psi(\beta, \alpha)\right],$$

given an observation X and hidden variables Z and θ. For the LDA model, we have:

$$p(X, Z, \theta|\beta, \alpha) = \prod_{n=1}^{N_X} p(x_n|z_n, \beta)p(z_n|\theta)p(\theta|\alpha) = \prod_{n=1}^{N_X}\prod_{k=1}^{K}\prod_{v=1}^{V}\left[(\beta_v^k)^{x_{nv}}\theta_k\right]^{z_{nk}} p(\theta|\alpha),$$

which we can expand as:

$$\begin{aligned} p(X, Z, \theta|\beta, \alpha) = \exp\left[\sum_{n=1}^{N_X}\sum_{k=1}^{K} z_{nk}\log\theta_k\right] \times \exp\left[\sum_{n=1}^{N_X}\sum_{k=1}^{K}\sum_{v=1}^{V} x_{nv}z_{nk}\log\beta_v^k\right] \\ \times \exp\left[\sum_{k=1}^{K}(\alpha_k - 1)\log\theta_k + B(\alpha)\right], \end{aligned}$$

with $B(\alpha) = \log\left[\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)\right] - \sum_{i=1}^{K}\log[\Gamma(\alpha_i)]$, where Γ is the gamma function. We deduce the non-canonical exponential family setting ϕ, S, ψ a:

$$S(X, Z, \theta) = \begin{pmatrix} S_{kv}^1 \equiv \left[\sum_{n=1}^{N_X} z_{nk}x_{nv}\right]_{kv} \\ S_k^2 \equiv [\log\theta_k]_k \end{pmatrix}, \tag{5}$$

$$\phi(\beta, \alpha) = \begin{pmatrix} \phi_{kv}^1 \equiv \left[\log\beta_v^k\right]_{kv} \\ \phi_k^2 \equiv [\alpha_k]_k \end{pmatrix}, \tag{6}$$

with $S^1, \phi^1 \in \mathbb{R}^{K\times V}$ and $S^2, \phi^2 \in \mathbb{R}^K$,

$$\psi(\beta, \alpha) = \sum_{i=1}^{K}\log[\Gamma(\alpha_i)] - \log\left[\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)\right], \tag{7}$$

and

$$a(X, Z, \theta) = \exp\left[\sum_{k=1}^{K}\left(\sum_{n=1}^{N_X} z_{nk} - 1\right)\log\theta_k\right].$$

The one-to one mapping between the sufficient statistics $s = \binom{s^1}{s^2}$ and (β, α) is defined by:

$$(\beta, \alpha)^*[s] = \begin{cases} \arg\max_{\beta\geq 0, \alpha\geq 0} & \langle\phi(\beta, \alpha), s\rangle - \psi(\beta, \alpha) \\ \text{s.t.} & \beta^\top\mathbf{1} = \mathbf{1}, \end{cases}$$

where $\mathbf{1}$ denotes the vector whose all entries equal 1. The objective function above $\langle\phi(\beta, \alpha), s\rangle - \psi(\beta, \alpha)$ is concave in β from the concavity of log and concave

in any α_k for $\alpha \geq 0$ as the function $B(\alpha)$ is concave as the negative log-partition of the Dirichlet distribution. We use the Lagrangian method for β:

$$L(\beta, \lambda) = \sum_{k=1}^{K} \sum_{v=1}^{V} s^1_{kv} \log \beta^k_v + \lambda^\top (\beta^\top \mathbf{1} - \mathbf{1}),$$

with $\lambda \in \mathbb{R}^K$. The derivative of L is set to zero when:

$$\forall (k, v), \frac{s^1_{kv}}{\beta^k_v} + \lambda_k = 0 \Rightarrow \lambda_k = -\sum_{v=1}^{V} s^1_{kv},$$

as $\sum_{v=1}^{V} \beta^k_v = 1$. We then have $(\beta^*(s))_{kv} = s^1_{kv} / \sum_j s^1_{kj}$. This mapping satisfies the constraint $\beta \geq 0$ because for any observation X and hidden variable Z, we have $S^1(X, Z)_{kv} \geq 0$. This comes from (5) and the fact that $\forall (n, k, v), (x_{nv}, z_{nk}) \in \{0, 1\}^2$. We find the condition on α by setting the derivatives to 0, which gives $\forall k \in [\![1, K]\!]$:

$$s^2_k - \Psi([\alpha^*(s)]_k) + \Psi\left(\sum_{i=1}^{K} [\alpha^*(s)]_i\right) = 0,$$

where $\Psi : x \mapsto \frac{\partial}{\partial x}[\log \Gamma](x)$ is the digamma function. Finally, $(\alpha^*(s), \beta^*(s))$ satisfies $\forall (k, v)$:

$$\begin{cases} (\beta^*(s))_{kv} & \equiv & \left[\frac{s^1_{kv}}{\sum_j s^1_{kj}}\right]_{kv} \\ \Psi([\alpha^*(s)]_k) - \Psi\left(\sum_{i=1}^{K} [\alpha^*(s)]_i\right) & = & s^2_k. \end{cases} \tag{8}$$

The parameter α^* is usually estimated with gradient ascent [Blei et al., 2003, Hoffman et al., 2010]. We can also estimate α with the fixed point iteration [Minka, 2000] which consists in repeating the following update until convergence:

$$\alpha^{new}_k = \Psi^{-1}\left(\Psi\left(\sum_{i=1}^{K} \alpha^{old}_i\right) + s^2_k\right).$$

We use the fixed point iteration to estimate α^* as it is more stable in practice. We study different updates for α in Appendix 3.8.

We can now apply Algorithm 1 to LDA. The only missing step is the estimation of the conditional expectation $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X, Z, \theta)]$, with $X = (x_1, \ldots, x_{N_X})$ and $Z = (z_1, \ldots, z_{N_X})$. We explain how to approximate this expectation with variational inference and Gibbs sampling.

3.3.2 Variational online EM applied to LDA (V-OEM)

In this section we explain how to approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X, Z, \theta)]$ with variational inference, in the frequentist setting. See Hoffman et al. [2013] for detailed derivations of variational inference for LDA in the Bayesian setting (from which

the updates in the frequentist setting may be easily obtained). The idea behind variational inference is to maximize the Evidence Lower BOund (ELBO), a lower bound on the probability of the observations:

$$p(X) \geq \text{ELBO}(X, p, q),$$

where q represents the variational model. In the case of LDA, the variational model is often set with a Dirichlet(γ) prior on θ and a multinomial prior on Z [Hoffman et al., 2013]:

$$q(Z, \theta) = q(\theta|\gamma) \prod_{n=1}^{N_X} q(z_n|\zeta_n). \tag{9}$$

We then maximize the ELBO with respect to γ and ζ, which is equivalent to minimizing the Kullback-Leibler (KL) divergence between the variational posterior and the true posterior:

$$\max_{\gamma,\zeta} \text{ELBO}(X, p, q) \Leftrightarrow \min_{\gamma,\zeta} \text{KL}[p(Z, \theta|X)||q(\theta, Z)]. \tag{10}$$

We solve this problem with block coordinate descent, which leads to iteratively updating γ and ζ as follows:

$$\zeta_{nk} \propto \prod_{v=1}^{V} \left(\beta_v^k\right)^{x_{nv}} \exp\left[\Psi(\gamma_k)\right], \tag{11}$$

$$\gamma_k = \alpha_k + \textstyle\sum_{n=1}^{N_X} \zeta_{nk}. \tag{12}$$

We then approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X, Z, \theta)]$ with the variational posterior. Given (5) and (9), we have:

$$\mathbb{E}_{p(Z,\theta|X)}[S(X, Z, \theta)] \approx \mathbb{E}_{q(Z,\theta)}[S(X, Z, \theta)] = \begin{bmatrix} \left(\sum_{n=1}^{N_{X_{t+1}}} \zeta_{nk} x_{nv}\right)_{kv} \\ \left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K} \gamma_j\right)\right)_k \end{bmatrix}. \tag{13}$$

The variational approximation of $\mathbb{E}_{p(Z,\theta|X)}[S(X, Z, \theta)]$ is then done in two steps:

1. Iteratively update ζ with (11) and γ with (12),
2. $\mathbb{E}_{p(Z,\theta|X)}[S(X, Z, \theta)] \leftarrow \mathbb{E}_{q(Z,\theta|\gamma,\zeta)}[S(X, Z, \theta)]$ with equation (13).

As γ and ζ are set to minimize the distance between the variational posterior and the true posterior (10) we expect that this approximation is close to the true expectation. However, as the variational model is a simplified version of the true model, there always remains a gap between the true posterior and the variational posterior.

3.3.3 Gibbs online EM applied to LDA (G-OEM)

In this section we explain how to approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X,Z,\theta)]$ with Gibbs sampling.

EXPECTATION OF S^1. Given (5), we have $\forall k \in [\![1,K]\!]$, $\forall v \in [\![1,V]\!]$:

$$\begin{aligned}\mathbb{E}_{Z,\theta|X,\alpha,\beta}\left[(S^1(X,Z))_{kv}\right] &= \mathbb{E}_{Z,\theta|X,\alpha,\beta}\left[\sum_{n=1}^{N_X} z_{nk}x_{nv}\right] \\ &= \sum_{n=1}^{N_X}\int_{Z,\theta} z_{nk}x_{nv}p(z_n,\theta|X,\beta,\alpha)\mathrm{d}\theta\mathrm{d}z \\ &= \sum_{n=1}^{N_X} x_{nv}p(z_{nk}=1|X,\beta,\alpha).\end{aligned}$$

We see that we only need the probability of z, and can thus use collapsed Gibbs sampling [Griffiths and Steyvers, 2004]. We have, following Bayes rule:

$$p(z_{nk}=1|z_{-n},X,\beta,\alpha) \propto p(x_n|z_{nk}=1,\beta)p(z_{nk}=1|z_{-n},\alpha),$$

where z_{-n} is the topic assignments except index n. In the LDA model, each word x_n is drawn from a multinomial with parameter β^{z_n}, which gives:

$$p(x_n|z_{nk}=1,\beta) = \sum_{v=1}^{V} x_{nv}\beta_v^k.$$

In the following, we use the notation $\beta_{x_n}^k \equiv \sum_{v=1}^{V} x_{nv}\beta_v^k$ for the sake of simplicty. We then use the fact that the topic proportions θ has a Dirichlet(α) prior, which implies that $Z|\alpha$ follows a Dirichlet-multinomial distribution (or multivariate Pólya distribution). As a result, the conditional distribution is:

$$p(z_{nk}=1|z_{-n},\alpha) = \frac{N_{-n,k}+\alpha_k}{(N_X-1)+\sum_j \alpha_j},$$

with $N_{-n,k}$ the number of words assigned to topic k in the current document, except index n. Finally, we have the following relation [Griffiths and Steyvers, 2004]:

$$p(z_{nk}=1|z_{-n},X,\beta,\alpha) \propto \beta_{x_n}^k \times \frac{N_{-n,k}+\alpha_k}{(N_X-1)+\sum_j \alpha_j}. \tag{14}$$

We estimate $p(z_{nk}=1|X,\beta,\alpha)$ with Gibbs sampling by iteratively sampling topic assignments z_n for each word, as detailed in Algorithm 2. We average over the last quarter of samples to reduce noise in the final output. We then incorporate the output in Algorithm 1.

EXPECTATION OF S^2. Given (5), we also have $\forall k \in [\![1, K]\!]$, $\forall v \in [\![1, V]\!]$:

$$\mathbb{E}_{Z,\theta|X,\alpha,\beta}[(S^2(X,Z))_k] = \mathbb{E}_{Z,\theta|X,\alpha,\beta}[\log \theta_k].$$

On the one hand, we have:

$$p(Z,\theta|X,\beta,\alpha) = p(Z|\theta,X,\beta,\alpha)p(\theta|X,\beta,\alpha) = C(\alpha)\prod_{k=1}^{K}\theta_k^{\left(\sum_{n=1}^{N_X} z_{nk}\right)+\alpha_k-1},$$

with $C(\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}$. On the other hand:

$$p(Z,\theta|X,\beta,\alpha) \propto p(\theta|Z,\alpha)p(Z|X,\beta).$$

We deduce from the two identities:

$$p(\theta|Z,\alpha) \propto \prod_{k=1}^{K}\theta_k^{\left(\sum_{n=1}^{N_X} z_{nk}\right)+\alpha_k-1} \Rightarrow \theta|Z,\alpha \sim \text{Dirichlet}\left(\alpha + \sum_{n=1}^{N_X} z_n\right).$$

Finally, the expectation is:

$$\begin{aligned}\mathbb{E}_{Z,\theta|X,\alpha,\beta}[(S^2(X,Z))_k] =& \mathbb{E}_{Z,\theta|X,\alpha,\beta}[\log\theta_k] \\ =& \mathbb{E}_{Z|X,\beta,\alpha}\left[\mathbb{E}_{\theta|Z,\alpha}\left[\log\theta_k\right]\right] \\ =& \mathbb{E}_{Z|X,\beta,\alpha}\left[\Psi\left([\alpha(s)]_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] - \Psi\left(\sum_{i=1}^{K}[\alpha(s)]_i + N_X\right),\end{aligned}$$

as the distribution of $\theta|Z,\alpha$ is Dirichlet $\left(\alpha + \sum_{n=1}^{N_X} z_n\right)$. We use the values of z sampled with Algorithm 2 to estimate this expectation. More precisely, keeping notations of Algorithm 2:

$$\mathbb{E}_{Z|X,\beta,\alpha}\left[\Psi\left([\alpha(s)]_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] \approx \frac{1}{P}\sum_{t=1}^{P}\Psi\left([\alpha(s)]_k + \sum_{n=1}^{N_X} z_{nk}^t\right).$$

3.3.4 Bayesian Approach

In a Bayesian setting, we consider β as a random variable, with $\beta \sim \text{Dirichlet}(b\mathbf{1})$, with $b \in \mathbb{R}$ and $\mathbf{1} \in \mathbb{R}^V$ denotes the vector whose all entries equal 1. The variational distribution of the global parameter β is set to $q(\beta^k|\lambda^k) = \text{Dirichlet}(\lambda^k)$, with $\lambda^k \in \mathbb{R}^V$ $\forall k = 1, \ldots, K$. The main difference with the frequentist methods above (`G-OEM` and `V-OEM`) is to optimize the ELBO with respect to the variational parameters $(\lambda^k)_k$. In practice, it is equivalent to replace β_v^k by $\exp\left[\mathbb{E}_q[\log\beta_v^k]\right]$ in all the updates above (i.e., in Equation (11) for `V-OEM` and in Equation (14) for `G-OEM`). The variational parmater $\lambda^k \in \mathbb{R}^V$ is updated with stochastic gradient on the ELBO, which gives, at iteration t:

$$\lambda^k(t+1) = \rho_t\lambda^k(t) + (1-\rho_t)\hat{\lambda}^k, \tag{15}$$

with $\hat{\lambda}^k \in \mathbb{R}^V$, $\hat{\lambda}_v^k = b + D\mathbb{E}_q\left[S_{kv}^1\right]$, where D is the total number of documents in the dataset and b is the prior on β^k [Hoffman et al., 2013].

Algorithm 2 Gibbs sampling scheme to approximate $p(z_{nk} = 1|X, \beta, \alpha)$

Input: β, α, X.

Initialization: $Z_n^0 \sim \text{Mult}([\bar{\beta}_{x_n}^k]_{k=1,\dots,K})$, with $\bar{\beta}_{x_n}^k = \frac{\beta_{x_n}^k}{\sum_j \beta_{x_n}^j}$ $\forall n \in [\![1, N_X]\!]$.

for $t = 1, 2, \dots, P$ **do**
  Compute random permutation σ^t on $[\![1, N_X]\!]$,
  for $n = 0, 1, \dots, N_X$ **do**
    • Set $Z_{-n}^t = \left\{(z_{\sigma^t(i)}^t)_{1 \leq i < n}, (z_{\sigma^t(i)}^{t-1})_{n < i \leq N_X}\right\}$,
    • Compute $\forall k$, $p(z_{\sigma^t(n)k} = 1|Z_{-n}^t, X, \beta, \alpha)$ with Equation (14),
    • Sample $z_{\sigma^t(n)}^t \sim \text{Mult}\left[p(z_{\sigma^t(n)}|Z_{-n}^t, X, \beta, \alpha)\right]$,
  end for
end for
for $n = 0, 1, \dots, N_X$ **do**
  Set $Z_{-n}^t = \left\{(z_i^t)_{1 \leq i < n}, (z_i^t)_{n < i \leq N_X}\right\}$ for $t \geq \frac{3}{4}P$,
  $p(z_{nk} = 1|X, \beta, \alpha) \leftarrow \frac{4}{P} \sum_{t=\frac{3}{4}P}^{P} p(z_{nk} = 1|Z_{-n}^t, X, \beta, \alpha)$
end for
Output: $\forall k$, $\forall n$: $(z_n^t)_{t=1,\dots,P}$, $p(z_{nk} = 1|X, \beta, \alpha)$.

3.4 Application to hierarchical Dirichlet process (HDP)

The HDP model [Teh et al., 2006] is a generative process to model documents from an infinite set of topics β_k, $k = 1, 2, 3, \dots$. Each topic is a discrete distribution of size V, the size of the vocabulary. Each topic is associated to a weight $\pi_k \in [0, 1]$, representing the importance of the topic in the corpus. For each document d, the (infinite) topic proportions ν_d are drawn from $\nu_d \sim \text{Dirichlet}(b\pi)$. We then generate words with a similar scheme to LDA scheme. More formally a corpus is generated as follows:

1. Draw an infinite number of topics $\beta_k \sim \text{Dirichlet}(\eta)$, for $k \in \{1, 2, 3, \dots\}$;
2. Draw corpus breaking proportions $\bar{\pi}_k \sim \text{Beta}(1, \alpha)$, for $k = 1, 2, 3, \dots$; with $\pi_k = \sigma_k(\bar{\pi})$;
3. For each document d:
   a) Draw document-level topic proportions: $\nu_d \sim \text{Dirichlet}(b\pi)$;
   b) For each word n in d:
      i. Draw topic assignment $z_{dn} \sim \text{Multinomial}(\nu_d)$;
      ii. Draw word $w_n \sim \text{Multinomial}(\beta_{z_{dn}})$.

In practice, we set the initial number of topics to $T = 2$. We then increase the number of topics used in the corpus using Gibbs sampling and the formula $p(z_{dn} > T|X, \eta) \propto b(1 - \sum_{i=1}^{T} \pi_i)$. See Section 3.4.2 for details.

3.4.1 HDP and Exponential Families

We consider an exponential family model on random variables (X, h) with parameter $\eta \in \mathcal{E} \subseteq \mathbb{R}^d$ and with density:

$$p(X, h|\eta) = a(X, h) \exp\left[\langle \phi(\eta), S(X, h)\rangle - \Psi(\eta)\right].$$

In the case of HDP, an observation X is a document of length N_X, where (as for LDA) $X = (x_1, \ldots, x_{N_X})$, $x_n \in \{0,1\}^V$ and $\sum_{v=1}^V x_{nv} = 1$. In the frequentist approach, the parameters of the model are global, represented by $\eta \equiv (\beta, \pi)$, where β represents the corpus topics, π represents the corpus breaking proportions. Our corpus $\mathcal{C}$ is a set of D observations $\mathcal{C} = (X_1, \ldots, X_D)$. For each document X^d, the associated hidden variables are $\nu_d \in [0,1]^K$ corresponding to document-level topic proportions. For each word x_n of document X^d, a hidden variable $z_n \in \{0,1\}^T$ is attached, corresponding to the topic assignment of word x_n.

We want to find ϕ, S, ψ and a such that, the joint probability is in the exponential family:

$$p(X, Z, \nu|\beta, \pi) = a(X, Z, \nu) \exp\left[\langle \phi(\beta, \pi), S(X, Z, \nu)\rangle - \psi(\beta, \pi)\right],$$

given an observation X and hidden variables Z and ν. For the HDP model, we have:

$$\begin{aligned}
p(X, Z, \nu|\beta, \pi) =& p(\nu|\pi) \prod_{n=1}^{N_X} p(x_n|z_n, \beta) p(z_n|\nu) \\
=& W(\pi) \prod_{k \in \mathbb{N}^*} (\nu_k)^{b\pi_k - 1} \prod_{n=1}^{N_X} \prod_k (\nu_k)^{z_{nk}} \prod_v (\beta_{k,v})^{x_{nv} z_{nk}} \\
=& \exp\left[-\psi(\pi)\right] \exp\left[\sum_k \log \nu_k \left(\sum_{n=1}^{N_X} z_{nk} - 1\right)\right] \\
& \times \exp\left[\sum_k (b\pi_k) \log \nu_k\right] \\
& \times \exp\left[\sum_{k,v} \log \beta_{k,v} \sum_{n=1}^{N_X} x_{nv} z_{nk}\right],
\end{aligned}$$

with $\psi(\pi) = \sum_k \log \Gamma(b\pi_k) - \log \Gamma(b)$.We deduce the exponential family setting ϕ, S, a:

$$S(X, Z, \nu) = \begin{pmatrix} S_k^1 \equiv [\log \nu_k]_k \\ S_{kv}^2 \equiv \left[\sum_{n=1}^{N_X} z_{nk} x_{nv}\right]_{kv} \end{pmatrix}, \tag{16}$$

$$\phi(\beta, \pi) = \begin{pmatrix} \phi_k^1 \equiv [b\pi_k]_k \\ \phi_{kv}^2 \equiv [\log \beta_{k,v}]_{kv} \end{pmatrix}, \tag{17}$$

with

$$a(X, Z, \nu) = \exp\left[\sum_k \log \nu_k \left(\sum_{n=1}^{N_X} z_{nk} - 1\right)\right].$$

The one-to one mapping between the sufficient statistics $s = (s^1, s^2)^\top$ and (β, π) is defined by:

$$(\beta, \pi)^*[s] = \begin{cases} \arg\max_{\beta \geq 0, 1 \geq \pi \geq 0} & \langle \phi(\beta, \pi), s \rangle - \psi(\beta, \pi) \\ \text{s.t.} & \beta^\top \mathbf{1} = \mathbf{1}, \end{cases}$$

where $\mathbf{1}$ denotes the vector whose all entries equal 1.

With the same computation than LDA, $\beta^*(s)_{kv} \equiv \left[\frac{s^2_{kv}}{\sum_j s^2_{kj}}\right]$. We find $\pi^*(s)$ by solving:

$$\pi^*(s) = \arg\max_{1 \geq \pi \geq 0} \sum_{k=1}^{K} \left(b\pi_k s^1_k - \log\Gamma(b\pi_k)\right) + \log\Gamma(b\sum_k \pi_k),$$

which gives:

$$\Psi(b\pi^*(s)_k) - \Psi\left(b\sum_i \pi^*(s)_i\right) = s^1_k.$$

where $\Psi : x \mapsto \frac{\partial}{\partial x}\left[\log\Gamma\right](x)$ is the digamma function. We estimate $(b\pi)^*$ with the fixed point iteration which consists in repeating the following update until convergence:

$$(b\pi)^{new}_k = \Psi^{-1}\left(\Psi\left(\sum_i (b\pi_i)^{old}\right) + s^1_k\right).$$

Finally, $(\beta, \pi)^*[s]$ satisfies $\forall(k, v)$:

$$\boxed{\begin{cases} (\beta^*(s))_{kv} & = & \left[\frac{s^2_{kv}}{\sum_j s^2_{kj}}\right] \\ \Psi(b\pi^*(s)_k) - \Psi\left(b\sum_i \pi^*(s)_i\right) & = & s^1_k. \end{cases}}$$

3.4.2 Inference with online EM

In this section, we explain how to approximate $\mathbb{E}_{Z,\nu|X,\eta}[S(X, Z, \nu)]$ with Gibbs sampling from a frequentist and a Bayesian perspective. In particular, as the total number of topics is infinite, we need to keep track of the previously used topics and iteratively extend the number of topics considered.

Gibbs online EM (`G-OEM`)

In our frequentist `G-OEM` approach, η is a parameter. The Gibbs sampling scheme to approximate $\mathbb{E}_{Z,\nu|X,\eta}[S(X, Z, \nu)]$ is different from LDA and a probability of adding a new topic to the current list is computed at each iteration, as explained below.

EXPECTATION OF S^1. We have:

$$\begin{aligned}\mathbb{E}_{Z,\nu|X,\eta}[S^1(X,Z,\nu)]_k &= \mathbb{E}_{Z,\nu|X,\eta}\left[\log \nu_k\right] \\ &= \mathbb{E}_{Z|X,\eta}\left[\Psi\left(b\pi_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] - \Psi\left(b\sum_i \pi_i + N_X\right),\end{aligned}$$

and we use the values of z sampled with Gibbs sampling to compute:

$$\mathbb{E}_{Z|X,\eta}\left[\Psi\left(b\pi_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] \approx \frac{1}{P}\sum_{t=1}^{P}\Psi\left(b\pi_k + \sum_{n=1}^{N_X} z_{nk}^t\right).$$

EXPECTATION OF S^2. We have:

$$\mathbb{E}_{Z,\nu|X,\eta}[S^2(X,\nu)]_{kv} = \mathbb{E}_{Z,\nu|X,\eta}\left[\sum_{n=1}^{N_X} z_{nk}x_{nv}\right] = \sum_{n=1}^{N_X} x_{nv}p(z_{nk}=1|X,\eta)$$

SAMPLING $z|X,\eta$. If T is the current number of topics, we have:

$$\begin{aligned}\forall k \in \{1,\dots,T\},\quad p(z_{nk}=1|z^{-n},X,\eta) &\propto (N_k^{-n} + b\pi_k) \times p(x_n|z_{ni}=1,c_{ik}=1,\eta) \\ &\propto (N_k^{-n} + b\pi_k) \times \beta_{k,x_n},\end{aligned}$$

and the probability of sampling a new topic is given by:

$$p(z_n > T|z^{-n},X,\eta) \propto b\left(1 - \sum_{t=1}^{T}\pi_k\right)/V.$$

When a new topic is generated, we initialize π_{T+1} with $\bar{\pi}_{T+1} \sim \text{Beta}(1,\alpha)$ and $\pi_{T+1} = \bar{\pi}_{T+1}\prod_{t=1}^{T}(1-\bar{\pi}_t)$.

Bayesian approach: `VarGibbs` *[Wang and Blei, 2012]*

In a Bayesian settings where $\beta^k \sim \text{Dirichlet}(\eta)$; $q(\beta^k|\lambda) = \text{Dirichlet}(\lambda^k)$ and $\pi_k \sim \text{Beta}(1,a)$; $q(\pi_k|a_k,b_k) = \text{Beta}(a_k,b_k)$, the sampling scheme is different as we also sample π and an auxiliary variable s_{dk} corresponding to the number of "tables" serving "dish" k in "restaurant" d (in the fomulation of HDP as a Chinese restaurant process; see Wang and Blei [2012] for details).
Sampling z:

$$p(z_{nk}=1|z^{-n},\lambda,\pi) \propto (N_{dk}^{-n} + b\pi_k)\frac{N_{kx_n}^{-n} + \lambda_{kx_n}}{N_k^{-n} + \sum_v \lambda_{kv}}.$$

Sampling s:

$$p(s_{dk}|N_{dk},b\pi_k) = \frac{\Gamma(b\pi_k)}{\Gamma(b\pi_k + N_{dk})}S(N_{dk},s_{dk})\,(b\pi_k)^{s_{dk}},$$

with $S(n, m)$ are unsigned Stirling number of the first kind.
Sampling π:

$$p(\bar{\pi}_k) \propto \bar{\pi}_k^{a_k - 1 + \sum_{d \in S} s_{dk}} \left(1 - \bar{\pi}_k\right)^{b_k - \alpha + \sum_{d \in S} \sum_{j=k+1}^{\infty} s_{dj}}$$

We then set:

$$\begin{cases} \hat{\lambda}_{kv} &= \eta + D \sum_{n=1}^{N_X} z_{nk} x_{nv} \\ \hat{a}_k &= 1 + D s_{dk} \\ \hat{b}_k &= \alpha + D \sum_{j=k+1}^{\infty} s_{dj} \end{cases} \tag{18}$$

and $\lambda^{t+1} = (1 - \rho_t)\lambda^t + \rho_t \hat{\lambda}$; $a^{t+1} = (1 - \rho_t)a^t + \rho_t \hat{a}$; $b^{t+1} = (1 - \rho_t)b^t + \rho_t \hat{b}$.

In practice, for each document we sample the hidden variables z for each word and compute the topic counts N_{dk} for topic k in document d, then we sample the variable s. Finally, we perform the online EM algorithm by making the approximation $\mathbb{E}_{p(h|X,\eta)}[S(X, h)] \approx \mathbb{E}_{q(h)}[S(X, h)]$, which corresponds to equation (18). Note that in this Bayesian approach, the parameters (λ, a, b) represent the distribution parameters of the random variables β and π.

3.5 EVALUATION

We evaluate our method by computing the likelihood on held-out documents, that is $p(X|\beta, \alpha)$ for any test document X. For LDA, the likelihood is intractable to compute. We approximate $p(X|\beta, \alpha)$ with the "left-to-right" evaluation algorithm [Wallach et al., 2009] applied to each test document. This algorithm is a mix of particle filtering and Gibbs sampling. On any experiments, this leads essentially to the same log-likelihood than Gibbs sampling with sufficiently enough samples—e.g., 200. In the following, we present results in terms of log-perplexity, defined as the opposite of the log-likelihood $-\log p(X|\eta)$. The lower the log-perplexity, the better the corresponding model. In our experiments, we compute the average test log-perplexity on N_t documents. We compare eight different methods:

- `G-OEM` (our main algorithm): Gibbs online EM. Online EM algorithm with Gibbs estimation of the conditional distribution $p(h|X, \eta)$ (Algorithm 2). Frequentist approach and step-size $\rho_i = 1/\sqrt{i}$;
- `V-OEM++`: variational online EM (also a new algorithm). Online EM algorithm with variational estimation of the conditional distribution $p(h|X, \eta)$, augmented with inference boosting from Section 3.2.3. Frequentist approach and step-size $\rho_i = 1/\sqrt{i}$;
- `OLDA`: online LDA [Hoffman et al., 2010]. Bayesian approach which maximizes the ELBO from Section 3.2.4, with natural stochastic gradient ascent and a step-size $\rho_i = 1/\sqrt{i}$;

- `VarGibbs`: Sparse stochastic inference for LDA [Mimno et al., 2012]. This method also maximizes the ELBO but estimates the variational expectations $q(Z, \theta)$ with Gibbs sampling instead of iterative maximization of variational parameters — see Section 3.3.2;

- `SVB`: streaming variational Bayes [Broderick et al., 2013]. A variational Bayesian equivalent of `V-OEM` with step-size $\rho_i = 1/i$;

- `SPLDA`: single pass LDA [Sato et al., 2010]. The difference with `V-OEM++` is that $\rho_i = 1/i$ and the updates in α done with a Gamma prior (see Appendix 3.8);

- `SGS`: streaming Gibbs sampling [Gao et al., 2016]. This method is related to `G-OEM` with $\rho_i = 1/i$. In this method, α is not optimized and set to a constant C_α. For comparison purposes, for each dataset, we set C_α to be the averaged final parameter $\hat{\alpha}$ obtained with `G-OEM` on the same dataset: $C_\alpha = \frac{1}{K}\sum_k \hat{\alpha}_k$. For each observation, only the last Gibbs sample is considered, leading to extra noise in the output;

- `LDS`: Stochastic gradient Riemannian Langevin dynamics sampler [Patterson and Teh, 2013]. The authors use the Langevin Monte Carlo methods on probability simplex and apply their online algorithm to LDA. For this method and only this method, we set to $P = 200$ the number of internal updates.

For existing variational methods—`OLDA`, `SVB`, `SPLDA`—β is a random variable with prior $q(\beta)$. We estimate the likelihood $p(X|\hat{\beta}, \alpha)$ with the "left-to-right" algorithm by setting $\hat{\beta} = \mathbb{E}_q[\beta]$ for Bayesian methods. For simplicity, we only present our results obtained with `G-OEM` and `V-OEM++`. Indeed, the inference boost presented in Section 3.3 is only beneficial for `V-OEM`. A detailed analysis is presented in Appendix 3.6.1.

3.5.1 Explicit links for LDA

In this section, we propose to make the links between the methods listed above explicit, using the framework described in Section 3.3 for the particular LDA model. We present in Table 4 a summary of the compared method.

CATEGORY: In the frequentist approach, β is a parameter and is updated with Equation (8), as the "M-step" in online EM.

In a Bayesian setting, β is a random variable with prior $\beta \sim \text{Dirichlet}(b\mathbf{1})$, with $b \in \mathbb{R}$ and $\mathbf{1} \in \mathbb{R}^V$ denotes the vector whose all entries equal 1. The variational distribution of the global parameter β is then set to $q(\beta^k|\lambda^k) = \text{Dirichlet}(\lambda^k)$, with $\lambda^k \in \mathbb{R}^V \ \forall k = 1, \ldots, K$. The variational parameter λ^k is updated by maximizing the ELBO with stochastic gradient ascent (Equation (15)).

Table 4: Comparison of existing methods for LDA.

	Category	$\mathbb{E}_{Z\|X,\eta}[S(X,Z)]$	Step-size ρ_t	Update for α
G-OEM	frequentist	Gibbs sampling	free	fixed point
V-OEM	frequentist	variational	free	fixed point
OLDA	Bayesian	variational	free	gradient ascent
VarGibbs	Bayesian	Gibbs sampling	free	α fixed
SVB	Bayesian	variational	fixed: $1/t$	gradient ascent
SPLDA	Bayesian	variational	fixed: $1/t$	Gamma prior
SGS	frequentist	Gibbs sampling	fixed: $1/t$	α fixed

estimation of $\mathbb{E}_{Z|X,\eta}[S(X,Z)]$: For LDA, the expectation $\mathbb{E}_{Z|X,\eta}[S(X,Z)]$ can either be estimated with Gibbs sampling—Equation (14)—or with variational approximation—Equation (10).

step-size: Some of the methods listed above (SVB,SPLDA and SGS) are incremental, which means the sufficient statistics are incrementally aggregated $s_t = s_{t-1} + \mathbb{E}_{Z_t|X_t,\eta}[S(X_t,Z_t)]$. For LDA, it exactly corresponds to a step-size $\rho_t = 1/t$ in the online EM setting, even though the link is not explicit in the corresponding papers.

For the other listed methods, the step-size exponent κ is chosen arbitrarily in $[0.5,1)$, with $\rho_t = 1/t^\kappa$. However, results are mostly presented with $\kappa = 1/2$ and $\rho_t = 1/\sqrt{t}$.

3.5.2 General settings

initialization. We initialize randomly $\eta \equiv (\beta,\alpha)$. For a given experiment, we initialize all the methods with the same values of (β,α) for fair comparison, except SPLDA that has its own initilization scheme—see Sato et al. [2010] for more details.

minibatch. We consider minibatches of size 100 documents for each update in order to reduce noise [Liang and Klein, 2009]. In the case of online EM in Equation (3), we estimate an expectation for each observation of the minibatch. We update the new sufficient statistics s towards the average of the expectations over the minibatch. We do the same averaging for all the presented methods.

number of local updates. For all the presented methods, we set the number of passes through each minibatch to $P = 20$. For G-OEM, this means that we perform 20 Gibbs sampling for each word of the minibatch. All other methods access each document 20 times (e.g., 20 iterations of variational inference on

Table 5: Datasets.

Dataset	#documents	$\overline{N_X}$	#words
Synthetic	1,000,000	60	1,000
Wikipedia[1]	1,010,000	162.3	7702
IMDB[2]	614,589	82.2	10,000
Amazon movies[3]	338,565	75.4	10,000
New York Times[4]	299,877	287.4	44,228
Pubmed[4]	2,100,000	82.0	113,568

each document). For `G-OEM`, inference with larger values for P (e.g., $P = 50$ or $P = 100$) leads to very similar results.

datasets. We apply the methods on six differents datasets, summarized in Table 5 ($\overline{N_X}$ is the average length of documents). Following Blei et al. [2003], the synthetic dataset has been generated from 10 topics and the length of each document drawn from a Poisson(60). The 10 topics are inferred with online LDA [Hoffman et al., 2010] from 50,000 reviews of the IMDB dataset with a vocabulary size of 10,000. We only consider the entries of the 1,000 most frequent words of this dataset that we normalize to satisfy the constraint $\sum_v \beta_v^k = 1$.

The words in the datasets IMDB, Wikipedia, New York Times, Pubmed and Amazon movies are filtered by removing the stop-words and we select the most frequent words of the datasets. For the synthetic dataset, IMDB, Pubmed and Amazon movies, the size of the test sets is $N_t = 5{,}000$ documents. For Wikipedia and New York Times, the test sets contain $N_t = 2{,}000$ documents. We run the methods on 11 differents train/test splits of each dataset. For all the presented results, we plot the median from the 11 experiments as a line—solid or dashed. For the sake of readability, we only present the same plots with error bars between the third and the seventh decile in Appendix 3.A and Appendix 3.B.

computation time. For each presented method and dataset, the computational time is reported in Table 6. Although all methods have the same running-time complexities, coded in Python, sampling methods (`G-OEM`, `VarGibbs` and `SGS`) need an actual loop over all documents while variational methods (`OLDA`, `SVB`, `SPLDA` and `V-OEM++`) may use vector operations, and may thus be up to twice faster. This could be mitigated by using efficient implementations of Gibbs sampling on minibatches [Yan et al., 2009, Zhao et al., 2014, Gao et al., 2016]. Note also that to attain a given log-likelihood, our method `G-OEM` is significantly faster

[1] Code available from Hoffman et al. [2010]
[2] Dataset described in Diao et al. [2014]
[3] Data from Leskovec and Krevl [2014]
[4] UCI dataset [Lichman, 2013]

Table 6: Average computational time (in hours) for each method — $K = 128$.

	IMDB	Wikipedia	NYT	Pubmed
G-OEM	13h	55h	30h	58h
V-OEM++	9h	37h	20h	54h
OLDA	7h	33h	8h	30h
VarGibbs	12h	50h	28h	54h
SVB	7h	34h	9h	30h
SPLDA	9h	37h	20h	54h
SGS	11h	48h	27h	50h
LDS	7h	17h	12h	40h

and often attains log-likelihoods not attainable by other methods (e.g., for the dataset New York Times).

step-size. In the following, we compare the results of our methods G-OEM and V-OEM++ with $\kappa = 1/2$, i.e., the step-size $\rho_t = 1/\sqrt{t}$, without averaging. Detailed analysis of different settings of our method can be found in Appendix 3.6. In particular, we compare different step-sizes and the effect of averaging over all iterates. We also compare the performance of OLDA with different step-sizes in Appendix 3.6.2 and observe that results are very similar for all the step-sizes that we try. Note that for incremental methods (SVB, SPLDA, SGS), the step-size is fixed to $\rho_t = 1/t$. For LDS, we run the method with parameters as close as possible to our method for fair comparison.

3.5.3 Results on LDA

Results obtained with the presented methods applied to LDA on different text datasets for different values of the number K of topics are presented in Figure 6. Performance through iterations (i.e., as the number of documents increases) is presented in Figure 9. We first observe that for all experiments, our new method G-OEM performs better—often significantly—than all existing methods. In particular, it is highly robust to diversity of datasets.

influence of the number of topics K. As shown in Figure 6, for synthetic data in plot (a), although the true number of topics is $K^* = 10$, SPLDA, OLDA, VarGibbs and SGS perform slightly better with $K = 20$, while G-OEM has the better fit for the correct value of K; moreover, SVB has very similar performances for any value of K, which highlights the fact that this method does not capture more information with a higher value of K. LDS performs very poorly on this dataset—for any value of K the log-perplexity is around 400—and is not displayed in Figure 6 (a) for clarity.

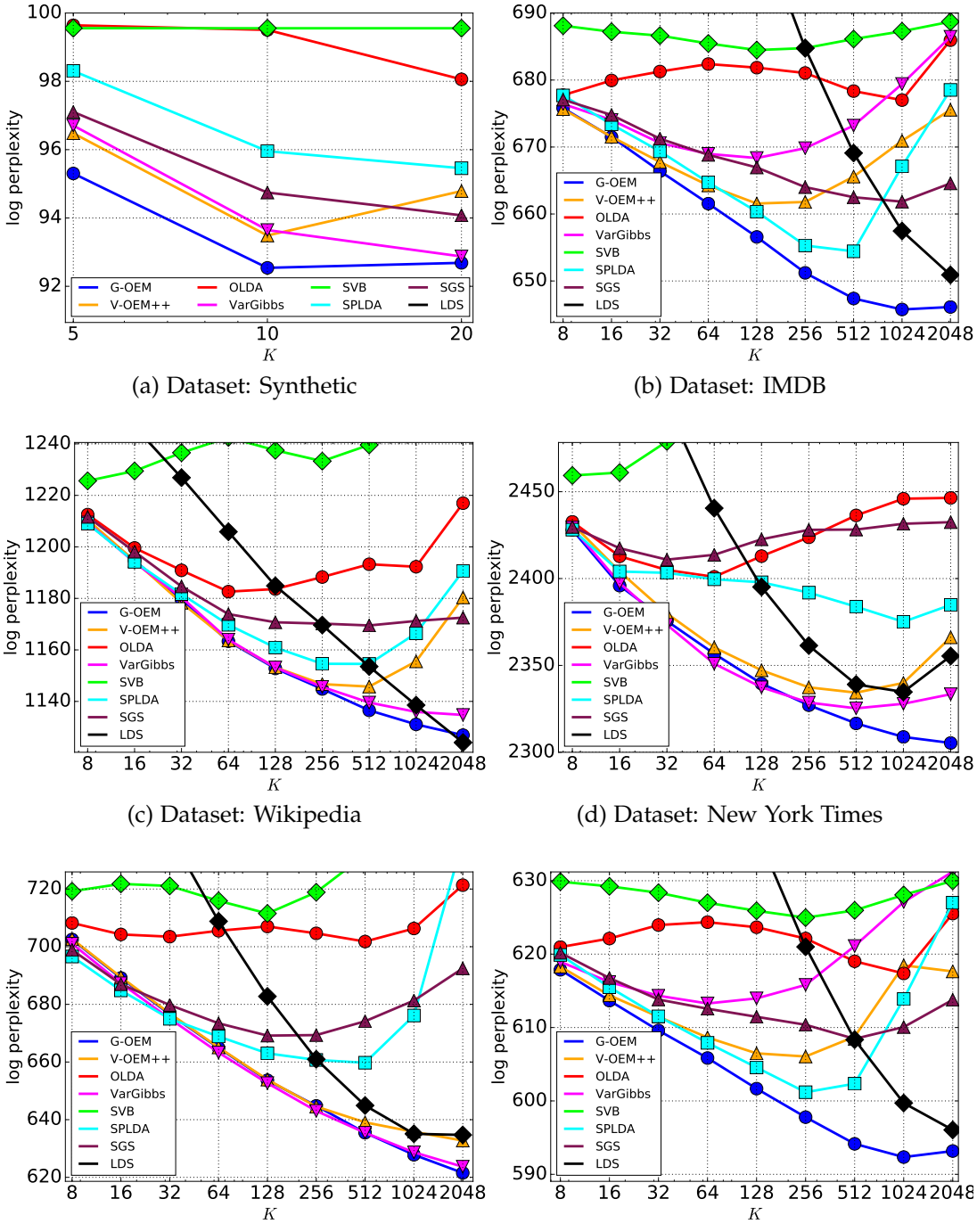
On non-synthetic datasets in plots (b)-(f), while the log-perplexity of frequentist methods—G-OEM, V-OEM++ and SPLDA—decreases with K, the log-perplexity of variational Bayesian methods—OLDA and SVB—does not decrease significantly with K. As explained below, our interpretation is that the actual maximization of the ELBO does not lead to an improvement in log-likelihood. The hybrid Bayesian method VarGibbs—which uses Gibbs sampling for local updates (θ, z) and variational updates for global parameters (β, α)—performs much better than the variational Bayesian methods. Our interpretation is that the objective function maximized with VarGibbs is a much better approximation of the log-likelihood than the ELBO.

In terms of robustness, G-OEM and LDS are the only methods that do not display overfitting on any dataset. However, LDS is only competitive for the highest values of K— $K \geq 500$.

PERFORMANCE THROUGH ITERATIONS. As shown in Figure 9, for synthetic data in plot (a), after only few dozens of iterations—few thousands of documents seen—G-OEM, V-OEM++ and VarGibbs outperform the other presented methods. Variational Bayesian methods again do converge but to a worse parameter value. On real datasets in plots (b)-(f), G-OEM and VarGibbs are significantly faster; we can indeed still observe that after around 100 iterations—10,000 documents seen—G-OEM and VarGibbs perform better than other methods on all the datasets except Pubmed, where the performances of G-OEM, V-OEM++, VarGibbs and SPLDA are similar.

VARIATIONAL VS. SAMPLING. Our method G-OEM directly optimizes the likelihood with a consistent approximation, and performs better than its variational counterparts SPLDA and V-OEM++ in all experiments. The hybrid method VarGibbs is less robust than G-OEM as it performs either similarly to G-OEM—for the datasets Wikipedia, New York Times and Pubmed—or worse than G-OEM and its variational counterparts SPLDA and V-OEM++—for the datasets IMDB and Amazon.

FREQUENTIST VS. BAYESIAN. In all our experiments we observe that frequentist methods—G-OEM, V-OEM++ and SPLDA—outperform variational Bayesian methods—OLDA and SVB. As described in Section 3.2.4, variational Bayesian methods maximize the ELBO, which makes additional strong independence assumptions and here leads to poor results. For example, as the number K of topics increases, the log-likelihood goes down for some datasets. In order to investigate if this is an issue of slow convergence, we show on Figure 7 (dotted black line) that running $P = 100$ internal updates in OLDA to get a finer estimate of the ELBO for each document may deteriorate the performance. Moreover, Figure 8 presents the evolution of the ELBO, which does always increase when K increases, showing that the online methods do optimize correctly the ELBO (while not improving the true log-likelihood). See Appendix 3.7 for additional results on the convergence of the ELBO. The results are mitigated for the hybrid



(a) Dataset: Synthetic

(b) Dataset: IMDB

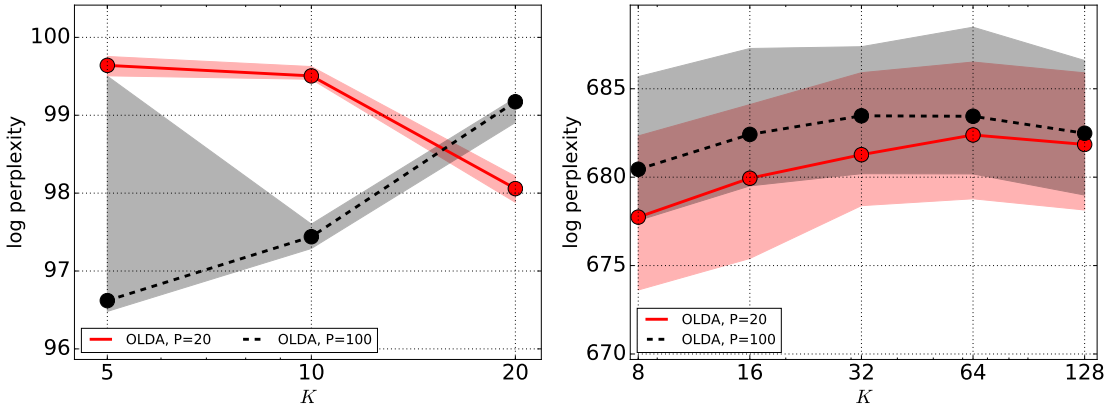
(c) Dataset: Wikipedia

(d) Dataset: New York Times

(e) Dataset: Pubmed

(f) Dataset: Amazon movies

Figure 6: Perplexity on different test sets as a function of K, the number of topics inferred. Best seen in color.



(a) Synthetic dataset. (b) IMDB.

Figure 7: OLDA. Perplexity on different test sets as a function of K for `OLDA` with $P = 10$ (red) and $P = 100$ (black) internal updates.

Bayesian method `VarGibbs`. The performance of this method is either similar to `G-OEM` and `V-OEM++` or significantly worse than both `G-OEM` and `V-OEM++`.

SMALL STEP-SIZES VS. LARGE STEP-SIZES. `SPLDA` is a variational method which is equivalent to `V-OEM++`, but with a step-size $1/i$, which is often slower than bigger step-sizes [Mairal, 2014], which we do observe—see Appendix 3.6.2 for a further analysis on the effect of the choice of step-sizes as $1/i^\kappa$ on `G-OEM`. Note that we run all the methods on a fixed (finite) number of observations. If we were to extend to infinite datasets, the difference between the step-sizes should be the speed of convergence. However, even if the number of observations is large, the gap between the step-sizes is still significant to justify the use of $1/\sqrt{t}$ for the step-size. Indeed, when considering large datasets, the contribution of each iteration at the end of the pass over the data is squeezed by the step-size in $1/t$. When the number of observations is large enough to prevent the use of batch algorithms but still insufficient for an online algorithm to converge in one pass, a possible solution could be to consider constant step-sizes in order to converge even faster to a local maxima. As proposed, we do not have any guarantee for our methods to converge with constant step-sizes, but previous works have shown the benefits of using constant step-sizes under certain assumptions (e.g., Bach and Moulines [2013])

3.5.4 Empirical analysis

In this section we provide a qualitative empirical analysis on the topics extracted with the different methods. We note this is clearly a subjective analysis but it stresses the benefits of a "better" inference mechanism in terms of log-likelihood

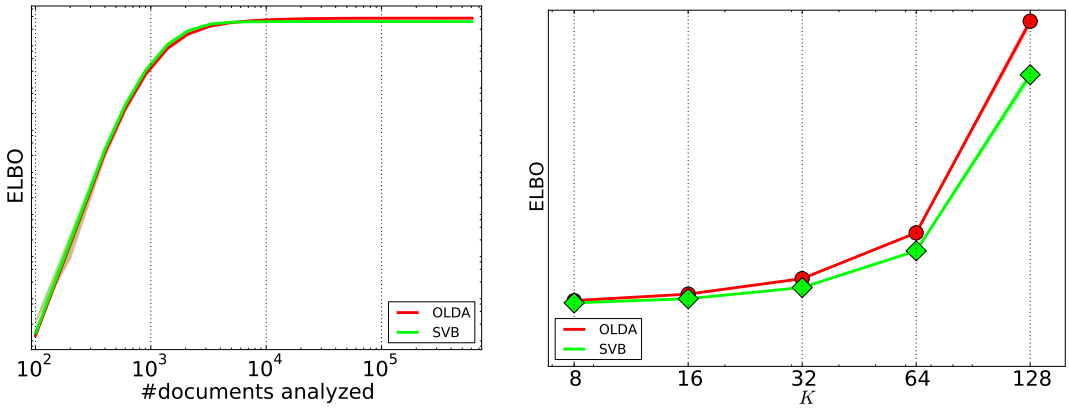


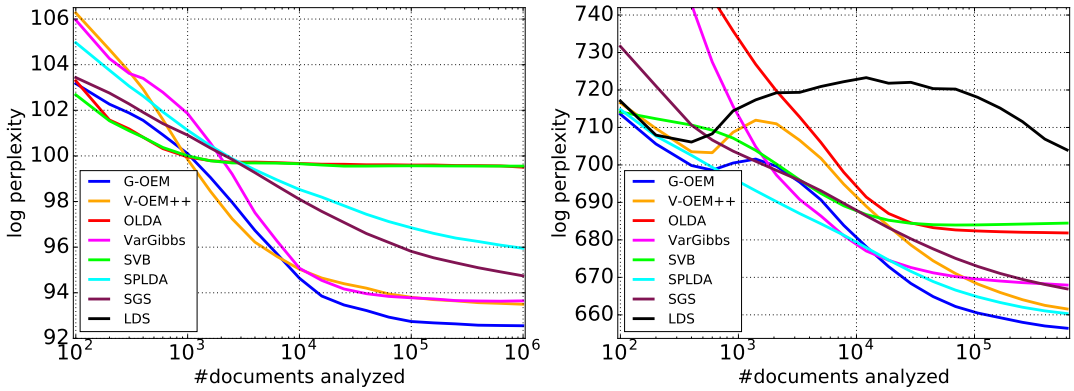
Figure 8: Dataset: IMDB. Evidence Lower BOund (ELBO) computed on test sets (20 internal iterations and 1 pass over the dataset). *Left*: ELBO through iterations with $K = 128$. *Right*: ELBO as a function of the number of topics K.

[Chang et al., 2009]. Examples of eight topics extracted with G-OEM and OLDA on the IMDB dataset of movie reviews are presented in Table 7 page 65.

We first compute the KL divergence between the $K = 128$ topics extracted with G-OEM and the $K = 128$ topics extracted with OLDA. We run the Hungarian algorithm on the resulting distance matrix to assign each topic extracted with G-OEM to a single topic of OLDA. We choose manually eight topics extracted with G-OEM that are representative of the usual behavior, and display the eight corresponding topics of OLDA assigned with the above method.

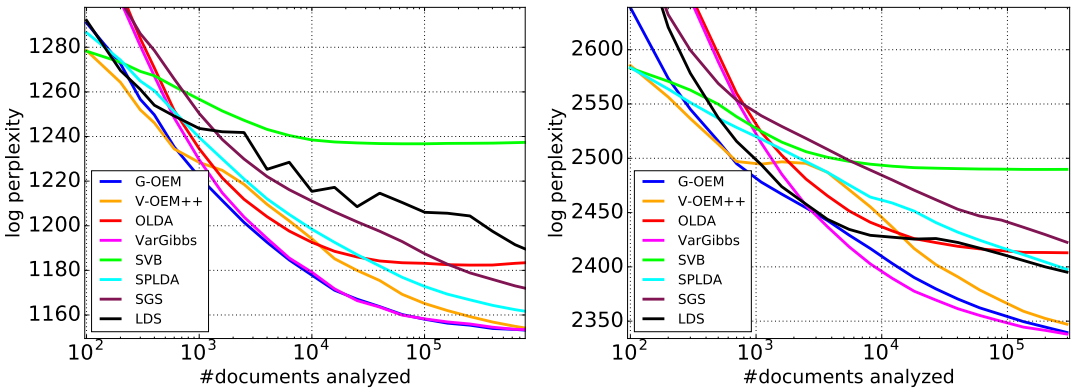
We observe that the topics extracted with G-OEM are more consistent than topics extracted with OLDA: topics of G-OEM precisely describe only one aspect of the reviews while the topics of OLDA tend to mix several aspects in each topic. For instance, the words of topic 1 extracted with G-OEM are related to *horror* movies. The words of the corresponding topic extracted with OLDA mix *horror* movies — e.g., *horror, scary* — and *ghost* movies — e.g., *ghost, haunt*. In this OLDA topic 1, we can also observe less relevant words, like *effective, mysterious*, which are not directly linked with *horror* and *ghost* vocabularies. We can make the same remarks with topic 2 and topic 3, respectively related to *comedy* movies and *romantic comedy* movies. In topic 2 extracted with G-OEM, the least related words to *comedy* are names of characters/actors — i.e., *steve* and *seth* — while the words not related to *comedy* in topic 25 of OLDA are more general, belonging to a different lexical field — e.g., *sport, site, progress, brave, definition*. In topic 3 of G-OEM, all the presented words are related to *romantic comedy* while in topic 3 of OLDA, the words *old, hard* and *review* are not related to this genre.

We also observe that G-OEM extracts strongly "qualitative" topics — topic 4 and topic 5 — which is not done with OLDA. Indeed, it is difficult to group the top words of topic 4 or topic 5 of OLDA in the same lexical field. Except *dialogue*



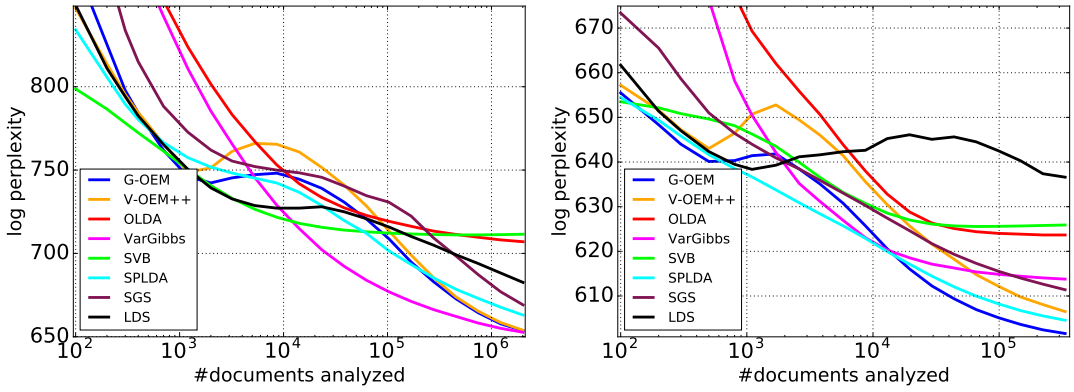
(a) Synthetic, $K = 10$

(b) IMDB, $K = 128$



(c) Wikipedia, $K = 128$

(d) New York Times, $K = 128$



(e) Pubmed, $K = 128$

(f) Amazon movies, $K = 128$

Figure 9: Perplexity through iterations on different test sets with the presented methods. Best seen in color.

and *suppose*, all the top words of topic 4 of G-OEM are negative words. These two words may appear in a lot of negative sentences, leading to a high weight in this topic. In topic 5 of G-OEM, the words *absolutely* and *visual* are non strictly positive words while the thirteen other words in this topic convey a positive opinion. The word *absolutely* is an adverb much more employed in positive sentences than negative or neutral sentences, which can explain its high weight in topic 5.

The topic 6 of both G-OEM and OLDA can be considered as a "junk" topic, as for both method, most of its top words are contractions of modal verbs or frequent words — e.g., *didn't, isn't, wait, bad*. The contractions are not filtered when removing the stop words as they are not included in the list of words removed[1].

For both G-OEM and OLDA, the top words of topic 7 are general words about movies. These words are usually employed to describe a movie as a whole — e.g., *narrative, filmmaker*.

Finally, the top words of topic 8 of G-OEM are related to the situation of the scenes. We could not find such topic in the other presented methods and we can see that the top words of topic 8 of OLDA — supposedly close to topic 8 of G-OEM — are related to *family* movies. Each word of topic 8 of G-OEM — except *group* and *beautiful* — are related to a *spatial location*, and may help answer the question "*where does the scene take place?*".

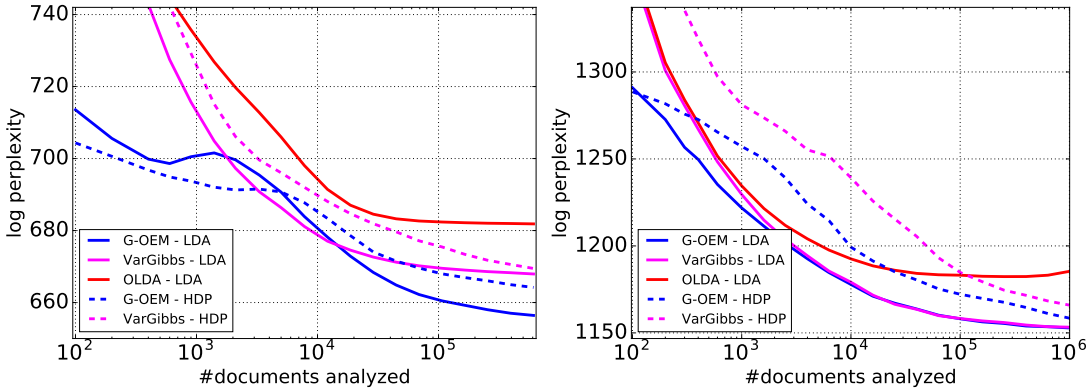
[1]See NLTK toolbox [Bird et al., 2009] for the exhaustive list of stop words.

Table 7: Comparison of topics extracted on IMDB dataset, $K = 128$ — 15 top words of eight topics extracted with G-OEM and OLDA.

				G-OEM				
#	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
1	violence	comedy	romantic	bad	brilliant	didn't	narrative	town
2	violent	funny	comedy	worst	perfect	i've	ultimately	local
3	disturbing	laugh	love	waste	beautiful	wasn't	cinematic	mountain
4	brutal	joke	romance	boring	masterpiece	isn't	approach	village
5	murder	hilarious	funny	awful	amazing	we're	protagonist	location
6	graphic	comic	charming	poor	superb	i'll	seemingly	road
7	killer	comedic	chemistry	dialogue	stunning	couldn't	nature	journey
8	torture	steve	sweet	worse	wonderful	wouldn't	tone	group
9	victim	amusing	enjoy	dull	absolutely	pretty	filmmaker	travel
10	rape	fun	heart	fail	best	bad	whose	country
11	kill	gag	nice	mess	incredible	haven't	craft	landscape
12	horror	seth	charm	ridiculous	brilliantly	guess	contemporary	land
13	bloody	funniest	great	suppose	beautifully	aren't	manner	beautiful
14	revenge	cameo	fun	terrible	visual	enjoy	serve	area
15	blood	situation	wonderful	unfortunately	perfectly	review	material	trip
				OLDA				
#	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
1	horror	hilarious	love	bad	great	wrong	visual	young
2	night	romance	enjoy	action	best	didn't	focus	family
3	dead	clever	pretty	interesting	star	i've	reality	child
4	twist	smart	old	original	long	wait	difficult	father
5	scary	intriguing	funny	far	john	catch	filmmaker	son
6	effective	comedic	comedy	special	excellent	exactly	image	age
7	mysterious	funniest	fun	fight	classic	wasn't	narrative	whose
8	bloody	progress	hard	hero	beautiful	huge	intelligent	tale
9	ghost	sport	perfect	entire	drama	i'll	accept	discover
10	haunt	site	laugh	half	wonderful	choice	impression	easily
11	fear	brave	entertaining	save	michael	etc	extreme	dream
12	evil	dreadful	worth	dialogue	heart	seriously	central	introduce
13	nightmare	shoulder	nice	full	forget	notice	maintain	marry
14	gory	gimmick	favorite	violence	robert	ridiculous	deny	raise
15	mask	definition	review	example	early	answer	nail	rule

3.5.5 Results on HDP

For the HDP model, we compare our G-OEM method to the Bayesian VarGibbs [Wang and Blei, 2012] method. We set the initial number of topics to $T = 2$. We present in Figure 10 results obtained with G-OEM and VarGibbs applied to both LDA and HDP. Results with error bars are presented in Appendix 3.C. For both LDA and HDP, G-OEM outperforms the Bayesian method VarGibbs.



(a) IMDB. (b) Wikipedia

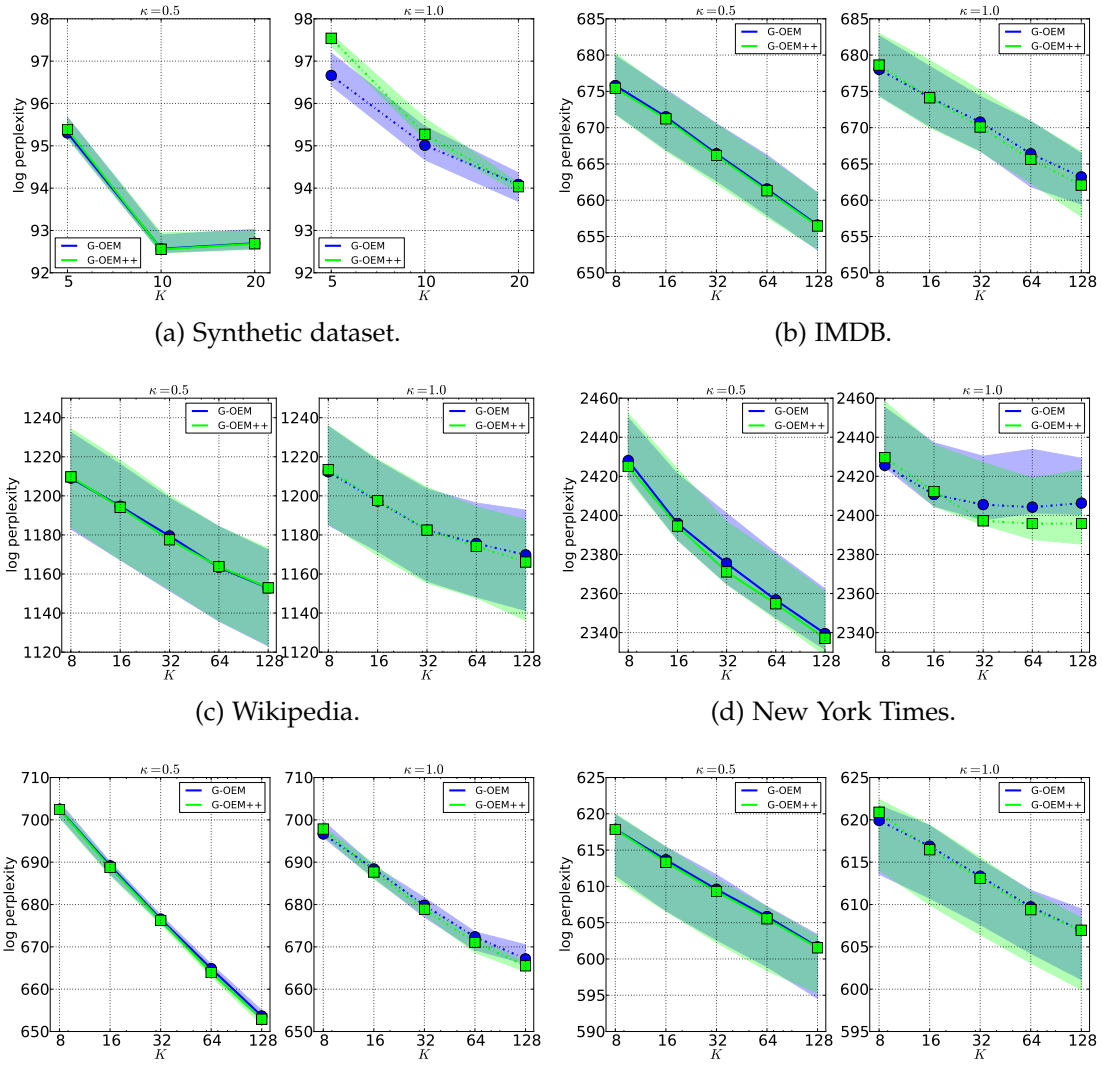
Figure 10: Perplexity through iterations on different test sets with G-OEM and VarGibbs applied to both LDA and HDP. Best seen in color.

3.6 GIBBS/VARIATIONAL ONLINE EM ANALYSIS

In this section we evaluate the proposed methods G-OEM and V-OEM with different settings in terms of step-sizes, averaging outputs and boosting internal updates.

3.6.1 Effect of inference boosting on G-OEM and V-OEM

The effect of the inference boost as described in Section 3.2.3 on G-OEM and V-OEM with synthetic and IMDB datasets is presented in Figure 11 and in Figure 12. It leads to a minor improvement for G-OEM++ and a significant one for V-OEM++.



(a) Synthetic dataset.

(b) IMDB.

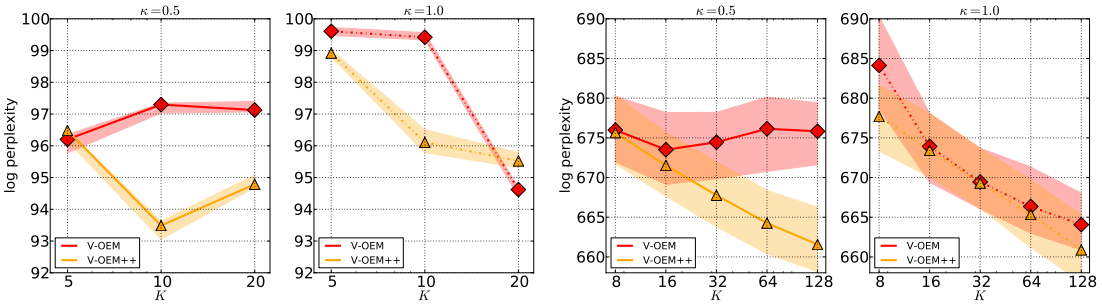
(c) Wikipedia.

(d) New York Times.

(e) Pubmed.

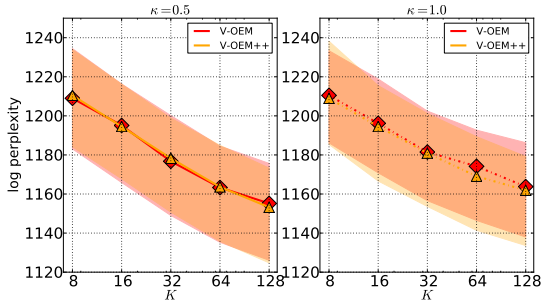
(f) Amazon movies.

Figure 11: G-OEM. Perplexity on different test sets as a function of the number of topics K for regular EM and boosted EM (++). We observe that for almost all datasets, there is no significant improvement when boosting the inference. Our interpretation is that each Gibbs sample is noisy and does not provide a stable boost. Best seen in color.

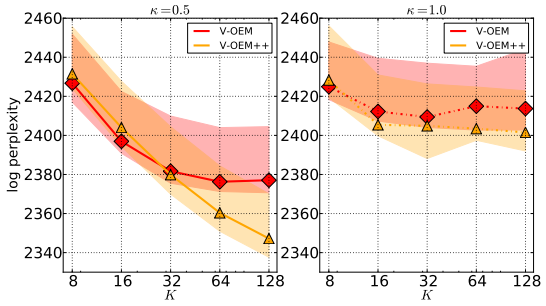


(a) Synthetic dataset.

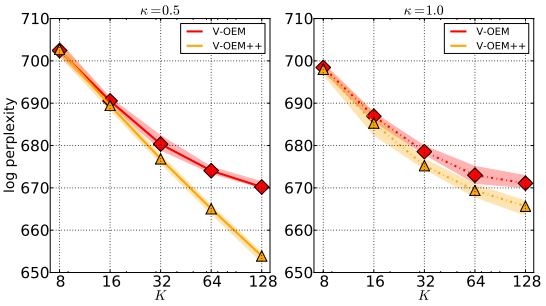
(b) IMDB dataset.



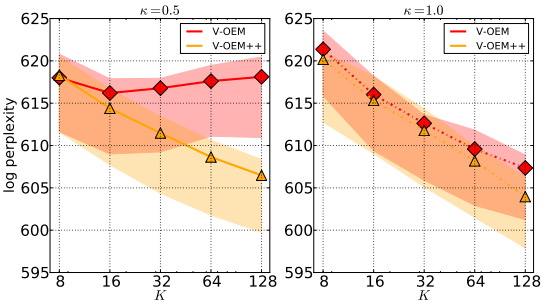
(c) Wikipedia.



(d) New York Times.



(e) Pubmed.



(f) Amazon movies.

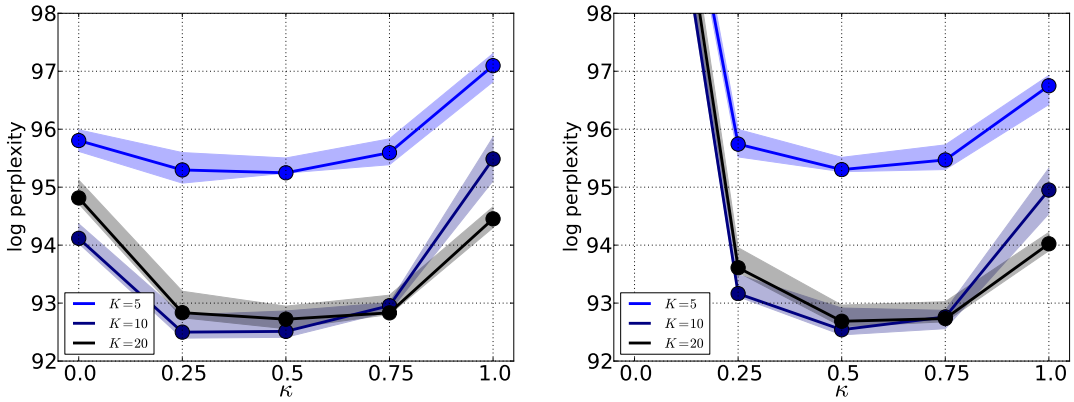
Figure 12: V-OEM. Perplexity on different test sets as a function of the number of topics K for regular EM and boosted EM (++). We observe that boosting inference improves significantly the results on all the datasets excepted on Wikipedia where V-OEM and V-OEM++ have similar performances. The variational estimation of the posterior is finer and finer through iterations. When updating the parameters at each iteration of the posterior estimation, the inference is indeed boosted. Best seen in color.

3.6.2 Step-sizes and averaging

We apply G-OEM with different stepsizes $\rho_i = \frac{1}{i^\kappa}$. Note that because we average sufficient statistics, there is no needed proportionality constants. We first compare the performance of the last iterate η_N (without averaging) and the average of the iterates $\bar{\eta}_N = \frac{1}{N}\sum_{i=0}^{N} \eta_i$ (with averaging) for different values of κ.

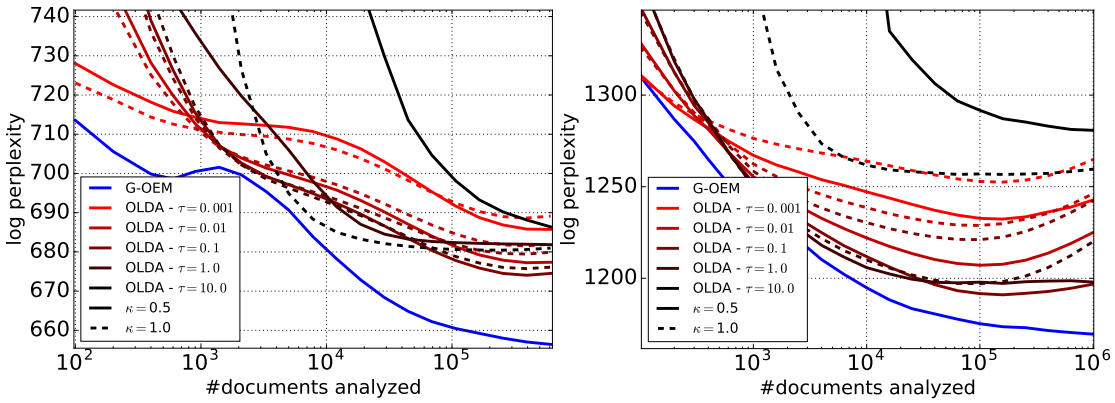
Results are presented in Figure 13 on the synthetic data and in Figure 16 on the IMDB dataset. For $\kappa \in \left[0, \frac{1}{2}\right[$, averaging improves the performance while for $\kappa \in \left]\frac{1}{2}, 1\right]$, averaging deteriorates the performance. For $\kappa = \frac{1}{2}$, averaging is only slightly beneficial on IMDB dataset. For constant stepsizes $\kappa = 0$ the averaging improves significantly the performance, as the iterates do not converge and tend to oscillate around a local optimum [Bach and Moulines, 2013]. We can expect the same effect for $\kappa \in \left[0, \frac{1}{2}\right[$ as the function $n \mapsto \frac{1}{n^\kappa}$ deacreases slowly for such values of κ. For $\kappa \in \left]\frac{1}{2}, 1\right]$, the stochastic gradient ascent scheme is guaranteed to converge to a local optimum [Bottou, 1998]. The averaging then deteriorates the performance as it incorporates the first iterates, which gets the last iterate away from local optimum. However, the stepsize $1/i$ ($\kappa = 1$) is not competitive. The performance with $\kappa = 0.75$ is only slightly better on IMDB dataset. The setting $\kappa = \frac{1}{2}$ represents a good balance between first and last iterates. For this stepsize, performances with or without averaging are similar but results without averaging seem to be more stable, hence our choice for all our other simulations.

We also apply `OLDA` with different step-sizes $\rho_t = \tau / t^\kappa$ for different values of τ, κ. Results are presented in Figure 14 without error bars and in Figure 15 with error bars. For `OLDA`, results are very similar for any step-size.



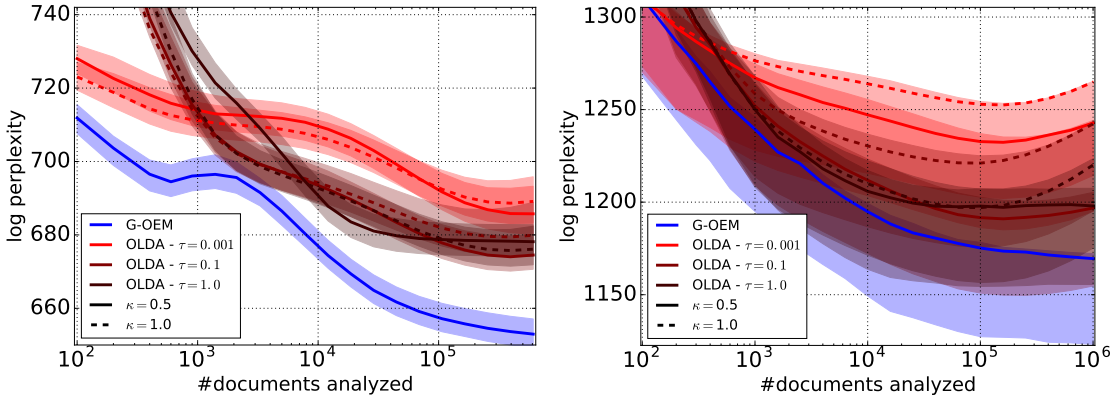
(a) *With* averaging (b) *Without* averaging

Figure 13: Dataset: synthetic. Perplexity on different test sets as a function of the exponent κ—the corresponding stepsize is $\rho_i = \frac{1}{i^\kappa}$—for `G-OEM` with averaging (left) and without averaging (right). The number of topics inferred K goes from 5 (the lightest) to 20 (the darkest). Best seen in color.



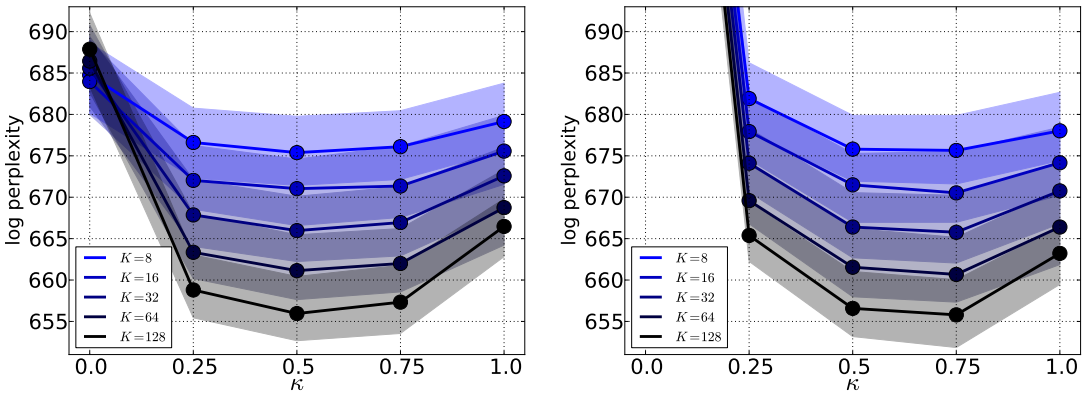
(a) IMDB. (b) Wikipedia.

Figure 14: Evolution of perplexity on different test sets as a function of the number of documents analyzed. For OLDA, we compare the performance with different step-sizes $\rho_t = \tau/t^\kappa$ for different values of τ, κ. Solid line: $\kappa = 1/2$; Dashed line: $\kappa = 1$.



(a) IMDB. (b) Wikipedia.

Figure 15: Evolution of perplexity on different test sets as a function of the number of documents analyzed, with error bars. For OLDA, we compare the performance with different step-sizes $\rho_t = \tau/t^\kappa$ for different values of τ, κ. Solid line: $\kappa = 1/2$; Dashed line: $\kappa = 1$.



(a) *With* averaging (b) *Without* averaging

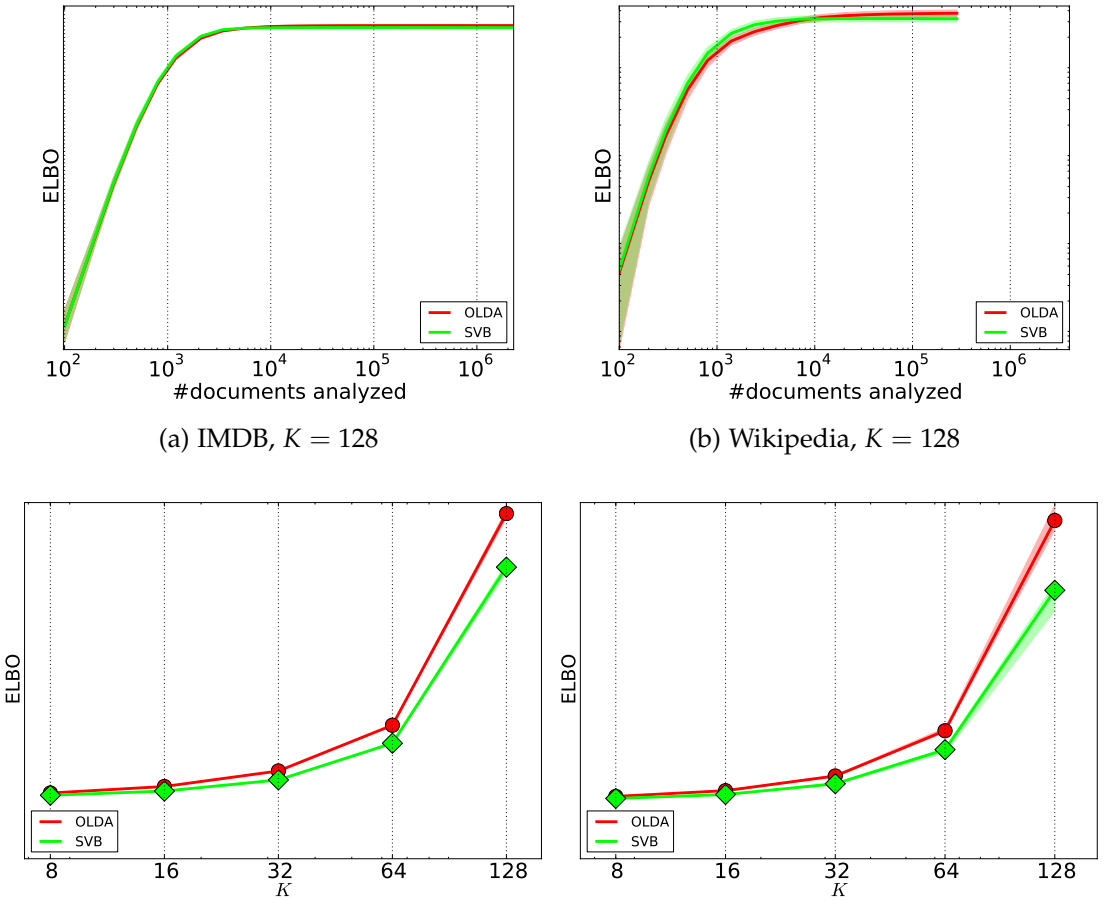
Figure 16: Dataset: IMDB. Perplexity on different test sets as a function of the exponent κ—the corresponding stepsize is $\rho_i = \frac{1}{i^\kappa}$—for `G-OEM` with averaging (left) and without averaging (right). The number of topics inferred K goes from 8 (the lightest) to 128 (the darkest). Best seen in color.

3.7 EVOLUTION OF THE ELBO

Figure 17 presents the evolution of the ELBO for online LDA (`OLDA`) and `SVB` on different test sets. We compute the ELBO on test documents as described by Hoffman et al. [2010]. This plot helps us to observe that even if the ELBO reaches a local maximum (i.e., it stabilizes), the quality of the model in terms of perplexity is not controllable. We can also see in Figure 17 that the ELBO is much better optimized with $K = 128$ than with other values of K for both `SVB` and `OLDA`, that is, as expected, latent variables of higher dimensionality lead to better fits for the cost function which is optimized. However, for several datasets the performance in terms of perplexity is better with low values of K ($K = 8$ or $K = 16$) than with high dimensional variables ($K = 64$ or $K = 128$).

Table 8: Comparison of log-perplexity levels reached with `OLDA` and `SVB` on IMDB dataset.

	$P = 200$, 4 passes	$P = 20$, 1 pass
OLDA	682.6±3.7	681.9±3.9
SVB	683.8±3.8	684.5±3.8



(a) IMDB, $K = 128$

(b) Wikipedia, $K = 128$

(c) IMDB

(d) Wikipedia

Figure 17: Evidence Lower BOund (ELBO) computed on different test sets. Top: ELBO through iterations, with 4 passes over each dataset and 200 internal iterations. Bottom: ELBO as a function of the number of topics K, with 20 internal iterations and 1 pass over each dataset. Best seen in color.

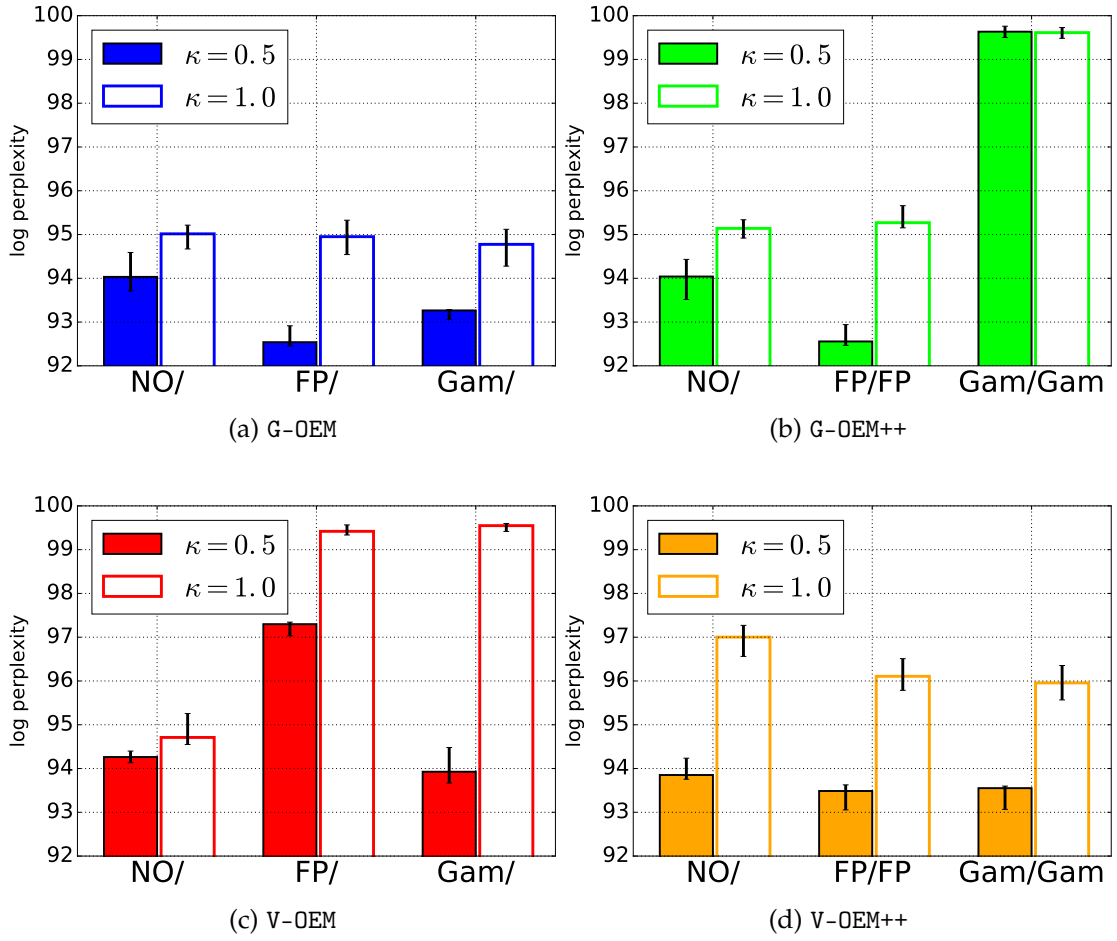
In order to check if more internal iterations could help variational Bayesian methods, we present in Table 8 the values of perplexity reached by `OLDA` when running 4 passes over each dataset with $P = 200$ internal iterations 1 pass over each dataset with $P = 20$ internal iterations. We observe that the ELBO converges quickly to a local optimum and doing ten times more internal iterations does not change significantly the final performance.

3.8 UPDATES IN α

In this section we compare the different types of updates for α. Figure 18 presents results obtained on synthetic dataset for fixed point iteration algorithm [Minka, 2000] and by putting a gamma prior on α [Sato et al., 2010]. We observe

that the fixed point method leads to better performance for G-OEM and G-OEM++. For V-OEM, the gamma updates better perform for $\kappa = \frac{1}{2}$. The performances of the gamma updates and the fixed point method are very similar for V-OEM++. Note that the algorithm V-OEM++ with $\kappa = 1$ and gamma updates on α is exactly equivalent to SPLDA [Sato et al., 2010]. The performance of this method can be improved by setting $\kappa = \frac{1}{2}$ with any update on α.

We also observe that fixing α to α_{true} that generated the data does not necessarily lead to better performance.



(a) G-OEM (b) G-OEM++

(c) V-OEM (d) V-OEM++

Figure 18: Dataset: Synthetic, $K = 10$. Perplexity on different test sets for different types of updates for α; for boosted methods, we use the same inference for α for local and global updates. NO: α is fixed and set to α_{true} that generated the data; FP: fixed point iteration; Gam: gamma prior on α [Sato et al., 2010]. Best seen in color.

3.9 CONCLUSION

We have developed an online inference scheme to handle intractable conditional distributions of latent variables, with a proper use of local Gibbs sampling within

online EM, that leads to significant improvements over variational methods and Bayesian estimation procedures. Note that all methods for the same problem are similar (in fact a few characters of code away from each other); ours is based on a proper stochastic approximation maximum likelihood framework and is empirically the most robust. It would be interesting to explore distributed large-scale settings [Broderick et al., 2013, Yan et al., 2009, Gao et al., 2016] and potentially larger (e.g., constant) step-sizes that have proved efficient in supervised learning [Bach and Moulines, 2013].

APPENDIX

3.A PERFORMANCE WITH DIFFERENT K, WITH ERROR BARS

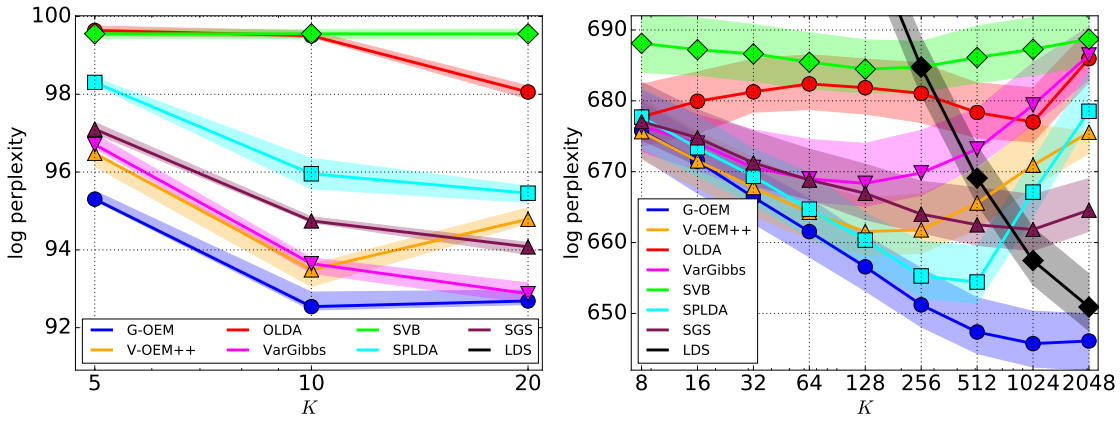
The performance of the presented methods for different values of K on the different datasets is presented in Figure 19. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.

3.B PERFORMANCE THROUGH ITERATIONS, WITH ERROR BARS

The performance through iterations of the presented methods on the different datasets is presented in Figure 20. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.

3.C RESULTS ON HDP, WITH ERROR BARS

The performance through iterations of the `G-OEM` and `VarGibss` applied to both LDA and HDP is presented in Figure 21. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.



(a) Dataset: Synthetic

(b) Dataset: IMDB

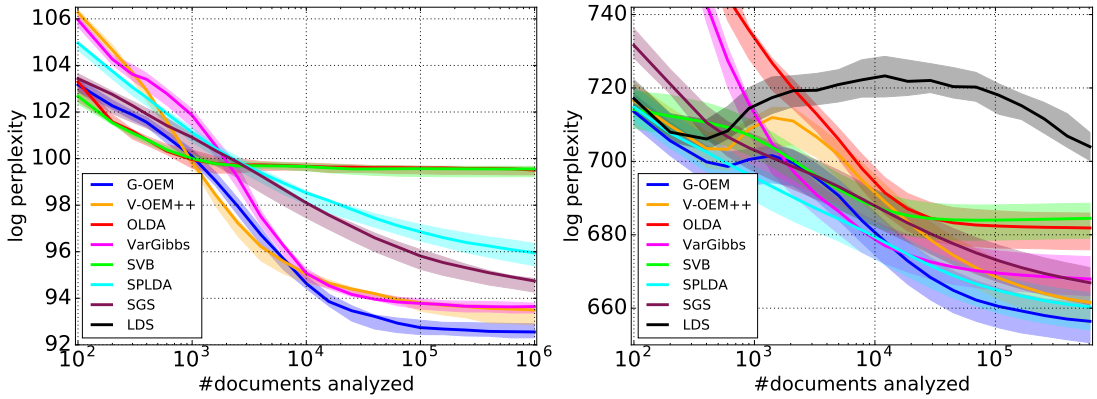
(c) Dataset: Wikipedia

(d) Dataset: New York Times

(e) Dataset: Pubmed

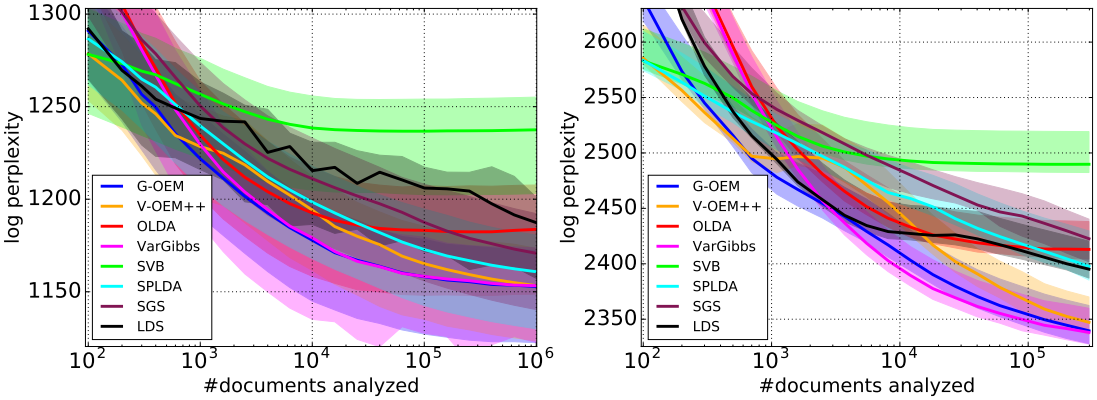
(f) Dataset: Amazon movies

Figure 19: Perplexity on different test sets as a function of K, the number of topics inferred. Same as Figure 6, but with error bars. Best seen in colors.



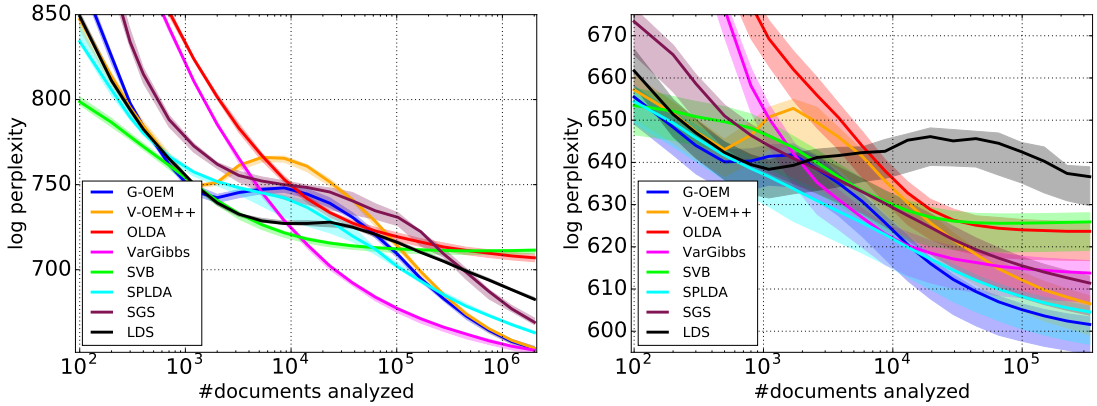
(a) Synthetic, $K = 10$

(b) IMDB, $K = 128$



(c) Wikipedia, $K = 128$

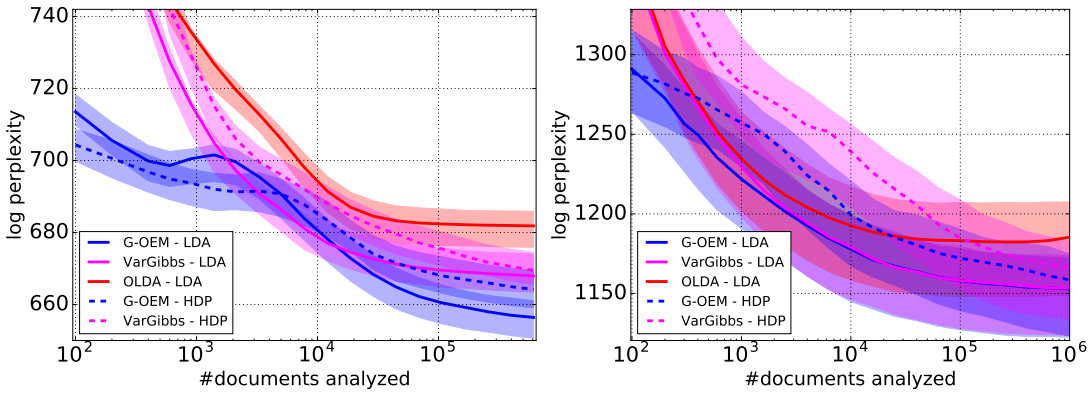
(d) New York Times, $K = 128$



(e) Pubmed, $K = 128$

(f) Amazon movies, $K = 128$

Figure 20: Perplexity through iterations on different test sets with the presented methods. Same as Figure 9, but with error bars. Best seen in colors.



(a) IMDB.

(b) Wikipedia

Figure 21: Perplexity through iterations on different test sets with `G-OEM` and `VarGibbs` applied to both LDA and HDP. Best seen in color.

4

DOCUMENT SUMMARIZATION WITH DETERMINANTAL POINT PROCESSES (DPP)

Determinantal point processes (DPPs) show lots of promises for modelling diversity in combinatorial problems, e.g., in recommender systems or text processing [Kulesza and Taskar, 2011, Gillenwater et al., 2012a, 2014], with algorithms for sampling [Kang, 2013, Affandi et al., 2013, Li et al., 2016a,b] and likelihood computations based on linear algebra [Mariet and Sra, 2015, Gartrell et al., 2016, Kulesza and Taskar, 2012].

While most of these algorithms have polynomial-time complexity, determinantal point processes are too slow in practice for large numbers N of items to choose a subset from. Simplest algorithms have cubic running-time complexity and do not scale well to more than $N = 1000$. Some progress has been made recently to reach quadratic or linear time complexity in N when imposing low-rank constraints, for both learning and inference [Mariet and Sra, 2016, Gartrell et al., 2016].

This is not enough, in particular for applications in continuous DPPs where the base set is infinite, and for modelling documents as a subset of all possible sentences: the number of sentences, even taken with a bag-of-word assumption, scales exponentially with the vocabulary size. Our goal in this chapter is to design a class of DPPs which can be manipulated (for inference and parameter learning) in potentially sublinear time in the number of items N.

In order to circumvent even linear-time complexity, we consider a novel class of DPPs which relies on a particular low-rank decomposition of the associated positive definite matrices. This corresponds to an embedding of the N potential items in a Euclidean space of dimension V. In order to allow efficient inference and learning, it turns out that a single operation on this embedding is needed, namely the computation of a second-order moment matrix, which would take time (at least) proportional to N if done naively, but may be available in closed form in several situations. This computational trick makes a striking parallel with positive definite kernel methods [Scholkopf and Smola, 2001, Shawe-Taylor and Cristianini, 2004], which use the "kernel trick" to work in very high dimension at the cost of computations in a smaller dimension.

In this chapter we make the following contributions:

- We propose in Section 4.2 a new class of determinantal point processes (DPPs) which is based on a particular low-rank factorization of the marginal kernel. Through the availability of a particular second-moment matrix, the complexity for inference and learning tasks is polynomial in the rank of the factorization and thus often sublinear in the total number of items (with exact likelihood computations).

– As shown in Section 4.3, these new DPPs are particularly suited to a subclass of continuous DPPs (infinite number of items), such as on $[0,1]^m$, and DPPs defined on the V-dimensional hypercube, which has 2^V elements.

– We propose in Section 4.4 a model of documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions. We present an application to document summarization with a DPP on 2^{500} items.

This work is under submission at ICML 2017.

4.1 REVIEW OF DETERMINANTAL POINT PROCESSES

In this chapter, for simplicity, we consider a very large *finite* set $\mathcal{X}$, with cardinality $|\mathcal{X}| = N$, following Kulesza and Taskar [2012]. In several places, we will consider an infinite set (see, e.g., Section 4.2.4) [Affandi et al., 2013, Lavancier et al., 2015].

A determinantal point process (DPP) on a set $\mathcal{X}$ is a probability measure on $2^{\mathcal{X}}$, the set of all subsets of $\mathcal{X}$. It can either be represented by an L-ensemble [Borodin and Rains, 2005] $L(x,y)$, for $x,y \in \mathcal{X}$ or by its marginal kernel $K(x,y)$, which we refer to as the "K-representation" and the "L-representation". In this chapter, K and L will be $N \times N$ matrices, with elements $K(x,y)$ and $L(x,y)$ for $x,y \in \mathcal{X}$. Both L and K are potentially large matrices, as they are indexed by elements of $\mathcal{X}$.

A sample X drawn from a DPP on $\mathcal{X}$ is a subset of $\mathcal{X}$, namely $X \subseteq \mathcal{X}$. In the "K-representation" of a DPP, for any set $A \subset \mathcal{X}$, we have:

$$\mathbb{P}(A \subseteq X) = \det K_A,$$

where K_A is the matrix of size $|A| \times |A|$ composed of pairwise evaluations of $K(x,y)$ for $x,y \in A$. If we denote by "$\preccurlyeq$" the positive semidefinite order on symmetric matrices (i.e., $A \preccurlyeq B \Leftrightarrow (B-A)$ is positive semidefinite), the constraint on K is $0 \preccurlyeq K \preccurlyeq I$ so that the DPP is a probability measure.

In the "L-representation", for any set $A \subset \mathcal{X}$, we have

$$\mathbb{P}(X = A) = \frac{\det L_A}{\det(I+L)}. \tag{19}$$

The constraint on L is $L \succcurlyeq 0$.

Given a DPP and its two representations L and K, we can go from L to K as $K = I - (I+L)^{-1}$ and vice-versa as $L = K(I-K)^{-1}$. The L-representation only exists when $K \prec I$, where "$\prec$" denotes the positive definite order of symmetric matrices (i.e., $A \prec B \Leftrightarrow (B-A)$ is positive definite). We denote by DPP(K,L) the DPP defined by the matrices (K,L) such that $L = K(I-K)^{-1}$.

Several tasks can be solved, e.g., marginalization, conditioning, etc., that are either easy in the L-representation or in the K-representation. For instance, (conditional) maximum likelihood when observing sets is easier in the L-representation,

as the likelihood of an observed set $A \subseteq \mathcal{X}$ is directly obtained with L through Eq. (19). Conversely, the expected number of selected items, $\mathbb{E}[|X|]$ for a DPP defined by L, K is easily computed with K as $\mathbb{E}[|X|] = \operatorname{tr} K$ [Kulesza and Taskar, 2012].

The DPPs model aversion between items. For instance, if X is drawn from a DPP(K, L), the probability that items i and j are together included in X is

$$\mathbb{P}(\{i, j\} \subseteq X) = K_{ii}K_{jj} - (K_{ij})^2.$$

This probability then decreases with similarity K_{ij} between item i and item j. This key aversion property makes DPPs useful to document summarization (see Section 4.4) where we want to select sentences that covers the most the document while avoiding redundancy.

APPROXIMATE COMPUTATIONS. In practice, the key difficulty is to deal with the cubic complexity in $|\mathcal{X}|$ of the main operations — determinant and computations of inverses. In their work, Kulesza and Taskar [2012] propose a low-rank model for the DPP matrix L, namely $L(x, y) = q(x)\langle \phi(x), \phi(y)\rangle q(y)$, where $q(x) \in \mathbb{R}^+$ corresponds to a "quality" measure of x and $\phi(x) \in \mathbb{R}^r$, $\|\phi(x)\| = 1$ corresponds to the "diversity" feature (or embedding) of x. In matrix notations, we have $L = \operatorname{Diag}(q)\Phi\Phi^\top \operatorname{Diag}(q)$. In particular, they show that most of the computations are based on the matrix $C = \Phi^\top \operatorname{Diag}(q^2)\Phi \in \mathbb{R}^{r\times r}$. As $\Phi \in \mathbb{R}^{N\times r}$, they achieve an overall complexity $O(Nr^2)$. In their application to document summarization, they only parameterize and learn the "quality" vector q, fixing the diversity features Φ, whereas we also parameterize and learn Φ.

More recently, Gartrell et al. [2016] use a low rank factorization of L ($L = UU^\top$, with $U \in \mathbb{R}^{N\times r}$) and apply accelerated stochastic gradient ascent on the log-likelihood of observed sets for learning U. They achieve a linear complexity in N: $O(Nr^2)$. Mariet and Sra [2016] propose a Kronecker factorization of L: $L = L_1 \otimes L_2$ where $\otimes$ is the Kronecker product, $L_i \in \mathbb{R}^{N_i\times N_i}$ and $N = N_1N_2$. They use a fixed point method (with the Picard iteration) to maximize the likelihood, that consists in alternatively updating L_1 and L_2 with a computational complexity $O(N^{3/2})$ if $N_1 \approx N_2 \approx \sqrt{N}$.

However, when the set $\mathcal{X}$ is very large (e.g., exponential) or infinite, even linear operations in $N = |\mathcal{X}|$ are intractable. In the next sections, we provide a representation of the matrices L and K together with an optimization scheme that makes the optimization of the likelihood tractable even when the set $\mathcal{X}$ is too large to perform linear operations in $N = |\mathcal{X}|$.

4.2 A TRACTABLE FAMILY OF KERNELS

We consider the family of matrices decomposed as a sum of the identity matrix plus a specific low-rank term, where the column-space of the low-rank term is fixed. We show that if K is in the family (with its additional constraint that $K \prec I$), so is L, and vice-versa.

4.2.1 Low-rank family

We consider a feature map $\phi : \mathcal{X} \to \mathbb{R}^V$, a probability mass function $p : \mathcal{X} \to \mathbb{R}_+$, a scalar $\sigma \in \mathbb{R}_+$ and a symmetric matrix $B \in \mathbb{R}^{V \times V}$. The kernel $K(x, y)$ is constrained to be of the form:

$$K(x, y) = \sigma 1_{x=y} + p(x)^{1/2} \phi(x)^\top B \phi(y) p(y)^{1/2}. \tag{20}$$

In matrix notation, this corresponds to

$$K = \sigma I + \operatorname{Diag}(p)^{1/2} \Phi B \Phi^\top \operatorname{Diag}(p)^{1/2},$$

with $\Phi \in \mathbb{R}^{|\mathcal{X}| \times V}$ and $B \in \mathbb{R}^{V \times V}$. With additional constraints detailed below, this defines a valid DPP, for which the L-representation can be easily derived in the form

$$L(x, y) = \alpha 1_{x=y} + p(x)^{1/2} \phi(x)^\top A \phi(y) p(y)^{1/2}, \tag{21}$$

with $\alpha \in \mathbb{R}_+$ and $A \in \mathbb{R}^{V \times V}$. The following proposition is a direct consequence of the Woodbury matrix identity:

Proposition 1 *The kernel K defined in Eq. (20) is a valid DPP if*
(a) $\sigma \in [0, 1]$,
(b) $0 \preccurlyeq B \preccurlyeq (1 - \sigma)(\Phi^\top \operatorname{Diag}(p) \Phi)^{-1}$.
It corresponds to the matrix L defined in Eq. (21) with $\sigma = \frac{\alpha}{\alpha+1}$ and

$$B = \frac{1}{(\alpha + 1)^2} \big[A^{-1} + \frac{1}{\alpha + 1} \Phi^\top \operatorname{Diag}(p) \Phi \big]^{-1}.$$

Moreover, we may use the matrix determinant lemma to obtain

$$\begin{aligned} \det(L + I) &= \det[(\alpha + 1) I] \det[A] \\ &\quad \times \det \Big(A^{-1} + \frac{1}{\alpha + 1} \Phi^\top \operatorname{Diag}(p) \Phi \Big), \end{aligned}$$

which is expressed through the determinant of a $V \times V$ matrix (instead of $N \times N$). We may also go from L to K as $\alpha = \frac{\sigma}{1 - \sigma}$ and

$$A = \frac{1}{(1 - \sigma)^2} \big[B^{-1} - \frac{1}{1 - \sigma} \Phi^\top \operatorname{Diag}(p) \Phi \big]^{-1}.$$

4.2.2 Tractability

From the identities above, we see that a sufficient condition for being able to perform the computation of L and its determinant is the availability of the matrix

$$\Sigma = \Phi^\top \operatorname{Diag}(p) \Phi = \sum_{x \in \mathcal{X}} p(x) \phi(x) \phi(x)^\top \in \mathbb{R}^{V \times V}.$$

In the general case, computing such an expectation would be (at least) linear time in N, but throughout this chapter, we assume this is available in polynomial

time in V (and not in N). As shown in Section 4.3, many standard distributions satisfy this property.

Note that this resembles the kernel trick, as we are able to work implicitly in a Euclidean space of dimension N while paying a cost proportional to V. In our document modelling example, we will have $N = 2^V$, and hence we achieve sublinear time.

We can compute other statistics of the DPP when K, L belong to the presented family of matrices. For instance, the expected size of a set X drawn from the DPP represented by K, L is:

$$\mathbb{E}[|X|] = \operatorname{tr} K = \sigma|\mathcal{X}| + \operatorname{tr}(B\Sigma). \tag{22}$$

Given $\phi(x)$, the parameters are the distribution $p(x)$ on $\mathcal{X}$, $A \in \mathbb{R}^{V\times V}$ and $\alpha \in \mathbb{R}_+$. If $\mathcal{X}$ is very large, it is hard to learn $p(x)$ from observations and $p(x)$ is thus assumed fixed. We also have to assume that α is proportional to $1/|\mathcal{X}|$ or zero when $\mathcal{X}$ is infinite as the first term of $\mathbb{E}[|X|]$ in Eq. (22), $\frac{\alpha}{\alpha+1}|\mathcal{X}|$, must be finite.

4.2.3 Additional low-rank approximation

If V is large, we use a low-rank representation for A:

$$A = \gamma I + U \operatorname{Diag}(\theta) U^\top, \tag{23}$$

with $\gamma \in \mathbb{R}_+$, $\theta \in \mathbb{R}_+^r$ and $U \in \mathbb{R}^{V\times r}$. All the exact operations on L are then linear in V, i.e., as $O(Vr^2)$. See details in Appendix 4.D. Moreover, the parameter θ can either be global or different for each observation, which gives flexibility to the model in the case where observations come from different but related DPPs on $\mathcal{X}$. For instance, in a corpus of documents, the distance between words (conveyed by U) may be different from one document to another (e.g., *field* and *goal* may be close in a sport context, not necessary in other contexts). This can be modelled through θ as topic proportions of a given document (see Section 4.4).

Note that the additional low-rank assumption (23) corresponds to an embedding $x \in \mathcal{X} \mapsto U^\top \phi(x) \in \mathbb{R}^r$, where $\phi(x)$ is fixed and U is learned. When $V \ll N$, the gain in complexity compared to $O(Nr^2)$ [Gartrell et al., 2016] or $O(N^{3/2})$ [Mariet and Sra, 2016] is significant. For instance, if we consider the ground set $\mathcal{X} = \{0,1\}^V$, we have $V = \log_2(N)$. The operations on L with our formulation are then sublinear in $N = 2^V$, i.e., as $O\left(\log_2(N)r^2\right)$. This complexity allows us to treat much larger problems $\left(\text{i.e., } V \gg 20 \text{ for } \mathcal{X} = \{0,1\}^V\right)$ that are intractable with previously achieved complexities.

4.2.4 Infinite $\mathcal{X}$

Although we avoid dealing rigorously with continuous-state DPPs in this work [Affandi et al., 2013, 2014, Bardenet and Titsias, 2015], we note that when dealing

with exponentially large finite sets $\mathcal{X}$ or infinite sets, we need to set $\sigma = \alpha = 0$ to avoid infinite (or too large) expectations for the numbers of sampled elements (which we use in experiments for $N = 2^V$).

Note moreover, that in this situation, the kernel K is formulated as

$$K(x, y) = p(x)^{1/2} p(y)^{1/2} \phi(x)^\top B \phi(y),$$

the rank of the matrix K is thus at most V, which implies that the number of sampled elements has to be less than V. This is not an issue in our experiments as V corresponds to the vocabulary size and we do not encounter documents with more than V sentences.

We can sample from the very large DPP as soon as we can sample from a distribution on $\mathcal{X}$ with density proportional to $p(x)\phi(x)^\top A\phi(x)$. Indeed, one can sample from a DPP by first selecting the eigenvectors of L, each with probability $\lambda_i/(\lambda_i + 1)$—where the λ_i's are the eigenvalues of L—and then projecting the canonical basis vectors—one per item—on this subset of eigenvectors. The density for selecting the first item is proportional to the squared norm of the latter projection (see Algorithm 1 of Kulesza and Taskar [2012] for more details). Given our fomula for L, all the required densities can be expressed as being proportional to $p(x)\phi(x)^\top A\phi(x)$. In our simulations, we use instead a discretized scheme.

4.2.5 Learning parameters with maximum likelihood

In this section, we present how to learn the parameters of the model, corresponding to the matrix A.

We have access to the likelihood through observations. We denote the observations by $X_1, \dots, X_M$, with $X_i \subseteq \mathcal{X}$, drawn from a density $\mu(X)$. Each set X_i is a set of elements $X_i = \{x_1^i, \dots, x_{|X_i|}^i\}$, with $x_j^i \in \mathcal{X}$. We denote by $\ell(X|L)$ the log-likelihood of a set X given a DPP matrix L. Our goal is to maximize the expected log-likelihood under μ, i.e., $\mathbb{E}_{\mu(X)}[\ell(X|L)]$. As we only have access to μ through observations, we maximize an estimation of $\mathbb{E}_{\mu(X)}[\ell(X|L)]$, i.e., $\mathcal{L}(L) = \frac{1}{M}\sum_{i=1}^M \ell(X_i|L)$. As the log-likelihood of a set $X \subseteq \mathcal{X}$ is given by $\ell(X|L) = \log\det L_X - \log\det(L + I)$, our objective function becomes:

$$\mathcal{L}(L) = \frac{1}{M}\sum_{i=1}^{M} \left(\log\det L_{X_i} - \log\det(I + L)\right). \tag{24}$$

In the following, we assume p fixed and we only learn A in its form (23).

In practice we minimize a penalized objective, that is, for our parameterization of A in Eq. (23),

$$F(L) = -\mathcal{L}(L) + \lambda\mathcal{R}(U, \theta),$$

where $\mathcal{L}$ is the log-likelihood of a train set of observations [Eq. (24)] and $\mathcal{R}$ is a penalty function. We choose the penalty $\mathcal{R}(U, \theta) = \|\theta\|_1 + \|U\|_{1,2}^2$ where $\|.\|_1$ is

the ℓ^1 norm and $\|U\|_{1,2} = \sum_{i=1}^r \|u_i\|_2$, where $\|.\|_2$ is the ℓ^2 norm and u_i is the i-th column of U. The group sparsity norm $\|.\|_{1,2}^2$ allows to set columns of U to zero and thus learn the number of columns.

This is a non non-convex problem made non smooth by the group norm. Following Lewis and Overton [2013], we use BFGS to reach a local optimum of our objective function.

4.3 EXAMPLES

In this section, we review our three main motivating examples: (a) orthonormal basis based expansions applicable to continuous space DPPs; (b) standard orthonomal embedding with $\mathcal{X} = \{1, \dots, N\}$ and (c) exponential set $\mathcal{X} = \{0,1\}^V$ for applications to document modelling based on sentences in Section 4.4.

4.3.1 Orthonormal basis based expansions

We consider a fixed probability distribution $p(x)$ on $\mathcal{X}$ and an orthonormal basis of the Euclidean space of square integrable (with respect to p) functions on $\mathcal{X}$. We consider $\phi(x)_i$ as the value at x of the i-th basis function. Note that this extends to any $\mathcal{X}$, even not finite by going to Hilbert spaces.

We consider $A = \mathrm{Diag}(a)$ and $B = \mathrm{Diag}(b)$ two diagonal matrices in $\mathbb{R}^{V \times V}$. Since $\phi(x)$ is an orthonormal basis, we have:

$$\Sigma = \sum_{x \in \mathcal{X}} p(x) \phi(x) \phi(x)^\top = I,$$

with a similar result for any subsampling of $\phi(x)$ (that is keeping a subset of the basis vectors).

For example, for $\mathcal{X} = [0,1]$, $p(x)$ the uniform distribution, $\alpha = 0$ and $\phi(x)$ the cosine/sine basis, we obtain the matrix $L(x,y) = \phi(x)^\top \mathrm{Diag}(a)\phi(y)$ which is a 1-periodic function of $x - y$, and we can thus model any of these functions. This extends to $\mathcal{X} = [0,1]^m$ by tensor products, and hyperspheres by using spherical harmonics [Atkinson and Han, 2012].

TRUNCATED FOURIER BASIS. In practice, we consider the truncated Fourier orthonormal basis of $\mathbb{R}^V$ with $V = 2d + 1$, i.e., $\phi_1(x) = 1$, $\phi_{2i}(x) = \sqrt{2}\cos(2\pi i x)$ and $\phi_{2i+1} = \sqrt{2}\sin(2\pi i x)$, for $i \in \{1, \dots, d\}$ and $x \in \mathcal{X}$. If $A = \mathrm{Diag}(a)$ is diagonal, then $L(x,y) = \phi(x)^\top \mathrm{Diag}(a)\phi(y)$ is a 1-periodic function of $x - y$, with only the first d frequencies, which allows us to learn covariance functions which are invariant by translation in the cube. We could also use the K-representation $K(x,y) = \phi(x)^\top B \phi(y)$, with $B = \mathrm{Diag}(b)$ diagonal in $[0,1]$, but the log-likelihood maximization is easier in the L-representation.

We use this truncated basis to optimize the log-likelihood $\mathcal{L}(L)$ on finite observations [Eq. (24)] , i.e., $X_i \subseteq \mathcal{X}$ and $|X_i| < \infty$. In particular, the normaliza-

tion constant is computed efficiently with this representation of L as we have $\det(L+I) = \prod_{i=1}^{V}(a_i+1)$.

NON-PARAMETRIC ESTIMATION OF THE STATIONARY COVARIANCE FUNCTION. We may learn any 1-periodic function of $x-y$ for $L(x,y)$ or $K(x,y)$ and we do so by choosing the truncated Fourier basis of size V, we could also use positive definite kernel techniques to perform non-parametric estimation.

RUNNING TIME COMPLEXITY. For general continuous ground set $\mathcal{X} = [0,1]^m$, with $m \geq 1$, the running time complexity is still controlled by $V = (2d+1)^m$, d corresponding to the number of selected frequencies in each dimension of the Fourier basis (with $a \in \mathbb{R}^V$). The value of d may be adjusted to fit the complexity in $O(V\kappa^3)$ or $O(d^m\kappa^3)$, where κ is the size of the biggest observation (i.e., the largest cardinality of all observed sets).

4.3.2 Standard orthonormal embeddings

In this section, we consider DPPs on the set $\mathcal{X} = \{1,\dots,V\}$ (i.e., $N = V$). We choose the standard orthonormal embedding, that is $\Phi = I$ which gives the expression $L = \alpha I + U\operatorname{Diag}(\theta)U^\top$, taking $\gamma = 0$ in Eq. (23). For this particular model, the complete embedding $\Phi U = U$ is learned and the distribution $p(x)$ is included in U. This is only possible when V is small. This model is suited to item selection, where groups of items are observed (e.g., shopping baskets) and we want to learn underlying embeddings of these items (through parameter U). Again, the size of the catalog V may be very large. Note that unlike existing methods leveraging low-rank representations of DPPs [Mariet and Sra, 2016, Gartrell et al., 2016], the parameter θ in our representation can be different for each observation, which makes our model more flexible.

4.3.3 $\mathcal{X} = \{0,1\}^V$

In this section, we consider DPPs on the set $\mathcal{X} = \{0,1\}^V$. For large values of V, direct operations on matrices L, K may be impossible as $\mathcal{X}$ is exponential, $|\mathcal{X}| = 2^V$. In particular, we consider the model where $\phi(x) = x$, i.e., $\Phi \in \mathbb{R}^{2^V \times V}$ (in other words we simply embed $\{0,1\}^V$ in $\mathbb{R}^V$).

As mentioned above, the tractability of the DPP(L,K) on $\mathcal{X} = \{0,1\}^V$ depends on the expectation $\Sigma = \sum_{x\in\mathcal{X}} p(x)xx^\top$. For particular distributions $p(x)$, Σ can be computed in closed form. For instance, if $p(x)$ corresponds to V independent Bernouillis, i.e., $p(x) = \prod_{i=1}^{V} \pi_i^{x_i}(1-\pi_i)^{1-x_i}$, the expectation quantity is $\Sigma = \operatorname{Diag}(\pi(1-\pi)) + \pi\pi^\top$. If the independent Bernoullis are exchangeable, i.e., all π's are equal, we have $\Sigma = \pi(1-\pi)I + \pi^2 11^\top$. Note that we can use the real numbers π_i as prior information, for instance by setting them to empirical frequencies of each word.

The tractability of our model is extended to the case $\mathcal{X} = \mathbb{N}^V$ with Poisson variables. Indeed, if $p(x) = \prod_{i=1}^{V} \frac{e^{-\lambda_i} \lambda_i^{x_i}}{x_i!}$ for $x \in \mathbb{N}^V$, the expectation of $xx^\top$ over $\mathcal{X}$ is $\Sigma = \mathrm{Diag}(\lambda) + \lambda\lambda^\top$.

Note that the tractability of the model is not restricted to these two examples of $p(x)$ (Bernouilli and Poisson) and can be easily extended to other distributions $p(x)$ (e.g., Gaussian) combined with other features $\phi(x)$. The distribution $p(x)$ corresponds to a prior knowledge on the considered items.

Given these examples, we assume in the following that:

$$\Sigma = \sum_{x \in \mathcal{X}} p(x) x x^\top = \mathrm{Diag}(\nu) + \mu\mu^\top.$$

The complexity of operations on the matrix L with this structure is $O(Vr^2)$—instead of $O(2^V r^2)$ if working directly with L; see Appendix 4.D for details. If we use the factorization of Σ above:

$$\begin{aligned} \mathrm{tr}\, K &= \frac{\alpha}{\alpha+1} 2^V + \frac{1}{(\alpha+1)^2} \mathrm{tr}\left[\left(\mathrm{Diag}(\nu) + \mu\mu^\top \right) \right. \\ &\quad \left. \times \left(\frac{1}{\alpha+1} (\mathrm{Diag}(\nu) + \mu\mu^\top) + A^{-1} \right)^{-1} \right]. \end{aligned}$$

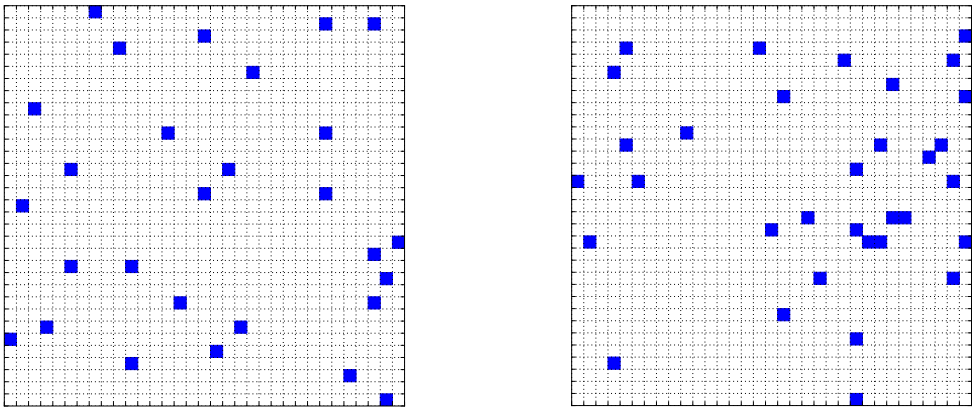
This identity suggests that we replace α by $\alpha 2^{-V}$ in order to select a finite set $X \subseteq \mathcal{X}$. For large values of V, $\alpha = 0$ is the key choice to avoid infinite number of selected items.

4.4 DPP FOR DOCUMENT SUMMARIZATION

We apply our DPP model to document summarization. Each document X is represented by its sentences, $X = (x_1, \ldots, x_{|X|})$ with $x_i \in \mathcal{X} = \{0,1\}^V$. The variable V represents the size of the vocabulary, i.e., the number of possible words. A sentence is then represented by the set of words it contains, ignoring their exact count and the order of the words. We want to extract the summary of each document as a subset of observed sentences. We use the structure described in Section 4.2.1 to build a generative model of documents. Let $K \in \mathbb{R}^{2^V \times 2^V}$ be the marginal kernel of a DPP on the possible sentences $\mathcal{X}$. We consider that the summary $Y \subseteq \mathcal{X}$ of document X is generated from the DPP(K, L) as follows:

1. Draw sentences $X = (x_1, \ldots, x_{|X|})$ from DPP represented by L,
2. Draw summary $Y \subseteq X$ from DPP represented by L_X.

In practice, we observe a set of documents and we want to infer the word embeddings U and the topic proportions θ for each document. In the following we consider that α and γ are fixed. We also denote by $\mathcal{L}(U, \theta) \equiv \mathcal{L}(L)$ the log-likelihood of observations [Eq. (24)] for simplicity as our DPP matrix L is encoded by U and θ.



(a) DPP. (b) I.i.d.

Figure 22: Comparison of points drawn from a DPP (left) independently from uniform distribution (right).

The intuition behind this generative model is that the sentences of a document cover a particular topic (the topic proportions are conveyed by the variable θ) and it is very unlikely to find sentences that have the same meaning in the same document. In this sense, we want to model aversion between sentences of a document.

PARAMETER LEARNING. As explained in Section 4.3.3, we assume that Σ is available in closed form, with $\Sigma = \text{Diag}(v) + \mu\mu^\top$. The log-likelihood of an observed document X is $\ell(X|L) = \log\det L_X - \log\det(L+I)$. The computation of the second term, $\log\det(L+I)$, is untractable to compute in reasonable time for any L when $V \geq 20$, since $L \in \mathbb{R}^{2^V \times 2^V}$. We can still compute this value exactly for structured L coming from our model with complexity $O(Vr^2)$ (see Appendix 4.D for details).

We infer the parameters U and θ by optimizing our regularized objective function $F(U,\theta) = -\mathcal{L}(U,\theta) + \lambda\mathcal{R}(U,\theta)$ with respect to U and θ alternatively. In practice, we perform 100 iterations of L-BFGS for the function $U \mapsto F(U,\theta)$ and 100 iterations of L-BFGS for each function $\theta_i \mapsto F(U,\theta)$, for $i = 1,\dots,M$. The optimization in U can also be done with stochastic gradient descent (SGD) [Bottou, 1998], using a mini-batch D_t of observations at iteration t: $U \leftarrow U - \rho_t G_t(U)$, with $G_t(U)$ the unbiased gradient:

$$G_t(U) = -\frac{1}{|D_t|}\sum_{i\in D_t} \nabla_U \ell(X_i|L(U,\theta_i)) + \lambda\nabla_U\mathcal{R}(U,\theta).$$

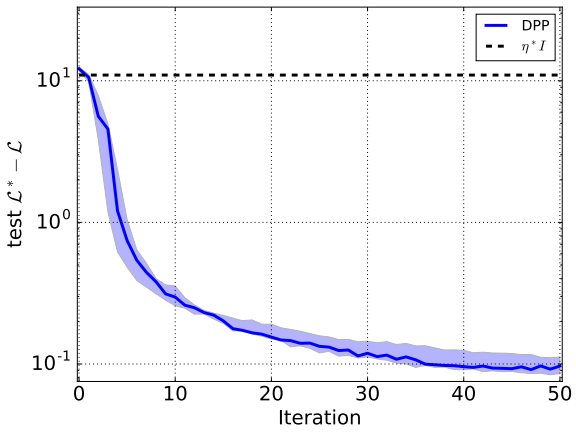
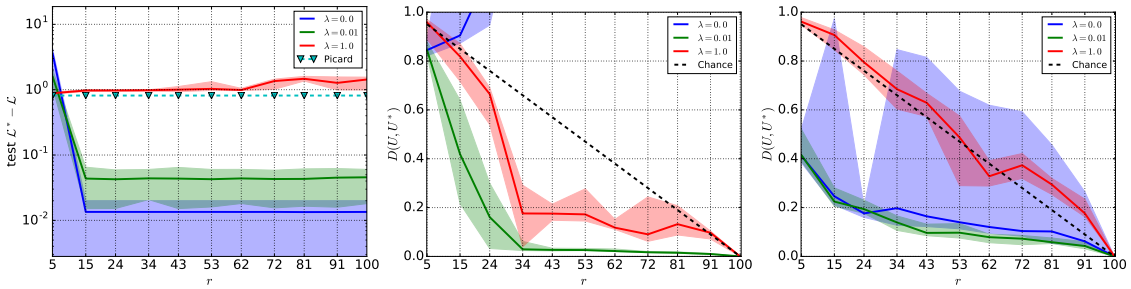


Figure 23: Continuous set $[0,1]^2$. Distance $(\mathcal{L}^* - \mathcal{L})$ in log-likelihood.



(a) Distance in log-likelihood. (b) Distance between U and U^*.(c) Distance between U and U^*.

Figure 24: Performance for ground set $\mathcal{X} = \{1, \ldots, V\}$ as a function of r. (a,b) Same θ for all the observations; (c) A different θ for each observation.

4.5 EXPERIMENTS

4.5.1 Datasets

We run experiments on synthetic datasets generated from the different types of DPPs described above. For all the datasets, we generate the observations using the sampling method described by Kulesza and Taskar [2012] (Algorithm 1 page 16) and perform the evaluation for 10 different datasets. This method draws exact samples from a DPP matrix L and its eigendecomposition (which requires N to be less than 1000). For the evaluation figures, the mean and the variance over the 10 datasets are respectively displayed as a line and a shaded area around the mean.

CONTINUOUS SET $[0,1]^m$. We describe in Section 4.3.1 a method to learn from subsets drawn from a DPP on a continuous set $\mathcal{X}$. As sampling from continuous DPPs is not straightforward and approximate [Affandi et al., 2013], we consider a discretization of the set $[0,1]^2$ into the discrete set $\{0, 1/N, \ldots, (N-1)/N\}^2$.

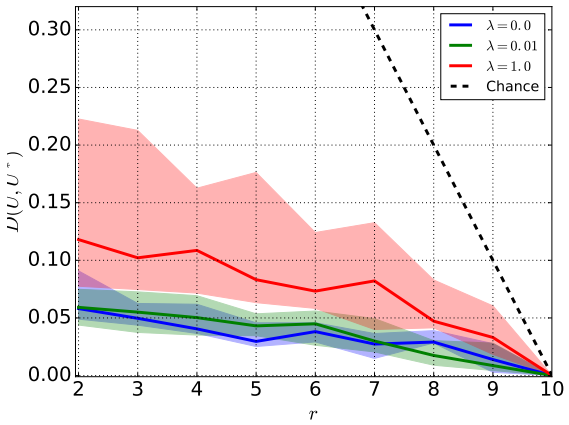


Figure 25: Performance for ground set $\mathcal{X} = \{0,1\}^V$ as a function of r with a different θ for each observation.

Note that this discretization only affects the sampling scheme. We generate a dataset from the ground set $\mathcal{X} = \{0, 1/N, \ldots, (N-1)/N\}^2$ with the DPP represented by $L(x,y) = \phi(x)^\top \mathrm{Diag}(a)\phi(y)$, with embedding $\phi(x) \in \mathbb{R}^{N^2}$ the discrete Fourier basis of $(\mathbb{R}^N)^2$, i.e., for $(i,j) \in \{1,\ldots,N\}^2$, $\phi(x)_{i,j} = \psi(x_1)_i \psi(x_2)_j$ with functions $\psi(z)_1 = 1$, $\psi(z)_{2k} = \sqrt{2}\cos(2\pi k z)$ and $\psi(z)_{2k+1} = \sqrt{2}\sin(2\pi k z)$, for $k = 1, \ldots, (N-1)/2$. With notations of Section 4.3.1 we have $V = N^2$. For $(i,j) \in \{0,\ldots,N-1\}^2$, we set $a_{(i,j)} = C_i C_j \tilde{a}_i \tilde{a}_j$, with $C_0 = 1$, $\tilde{a}_0 = 1$ and $C_i = 1/\sqrt{2}$, $\tilde{a}_i = 1/i^\beta$ for $i \geq 1$. We choose $N = 33$ (i.e., $V = N^2 = 1069$) and $\beta = 2$ for the experiments. We present in Figure 22 two samples: a sample drawn from the DPP described above and a set of points that are i.i.d. samples from the uniform distribution on $\mathcal{X}$. We observe aversion between points of the DPP sample that are distributed more uniformly than points of the i.i.d. samples.

ITEMS SET. We generate observations from the ground set $\mathcal{X} = \{1,\ldots,V\}$, which corresponds to the matrix $L = \alpha I + U\,\mathrm{Diag}(\theta)U^\top$. For these observations, we set $V = 100$, $r = 5$, $\alpha = 10^{-5}$. For each dataset, we generate U and θ randomly with different seeds accross the datasets.

EXPONENTIAL SET. We generate observations from the set $\mathcal{X} = \{0,1\}^V$ with $\phi(x) = x$. In this case, we set $V = 10$, $r = 2$, $\alpha = 10^{-5}$, $\gamma = 1/V$. As we need the eigendecomposition of $L \in \mathbb{R}^{2^V \times 2^V}$ for sampling, we could not generate exact samples with higher orders of magnitude for V. However, we can still optimize the likelihood for ground sets with large values of V and we run experiments on real document datasets, where the size of the vocabulary is $V = 500$ (i.e., $|\mathcal{X}| = 2^{500} \approx 10^{150}$).

For both ground sets $\mathcal{X} = \{1,\ldots V\}$ and $\mathcal{X} = \{0,1\}^N$, we consider two types of datasets: one dataset where all the observations are generated with the same DPP matrix L and another dataset where observations are generated with a

Table 9: Examples of reviews with extracted summaries (of size $l = 5$ sentences) colored in blue.

Review 1
Ate here once each for dinner and Sunday brunch. **[Dinner was great.] [We got a good booth seat and had some tasty food.]** I ordered just an entree since I wasn't too hungry. The guys ordered appetizers and salad and I couldn't resist trying some. The risotto with rabbit meatballs was so good. **[Corn soup, good.] [And my duck breast, also good.]** I was happy. **[The sides were good too.]** Potatoes and asparagus. Came back for Mother's Day brunch. Â Excellent booth table at the window, so we could watch our valeted car. Pretty good service. Good food. No complaints.
Review 2
This will be my 19 month old's first bar. :D I came here with a good friend and my little guy. We shared the double pork chop and the Mac n Cheese. **[The double pork chop was delicious.....] [Huge portions and beautifully prepared vegetables.] [What a wonderful selection of butternut squash, spinach, cauliflower and mashed potato.]** We were very impressed with the chop, meat was tender and full of flavor. **[The mac n cheese, was okay.]** I would definitely go back for the pork chop... might want to try the fried mushrooms too. **[Place surprisingly was pretty kid friendly.]** The bathroom actually had a bench I could change my little guy!

different matrix $L(\theta^i)$ for each observation. For the second type of dataset, the embedding U is common to all the observations while the variable θ^i differs from one observation to another.

REAL DATASET. We consider a dataset of 100,000 restaurant reviews and minimize the objective function $F(U, \theta)$ mentioned above. We first remove the stopwords using the NLTK toolbox [Bird et al., 2009]. Among the remaining words, we only keep the $V = 500$ most frequent words of the dataset. After filtering, the average number of sentences per review is 10.5 and each sentence contains on average 4.5 words. We use the proposed DPP structure to (1) learn word embedding U from observations and (2) extract a summary for each review using the model of Section 4.4. Given a document X, the inferred parameters U and $\theta(X)$ and the corresponding DPP matrix L, we extract the l sentences summarizing the document X by solving the following maximization:

$$Y^* \in \arg \max_{Y \subseteq X,\, |Y|=l} \frac{\det(L_Y)}{\det(L_X + I)}.$$

In practice we use the greedy MAP algorithm [Gillenwater et al., 2012b] to extract the summary $\widehat{Y}$ of document X, as an approximation of the MAP Y^*

with the usual submodular maximization approximation guarantee [Krause and Golovin, 2012].

4.5.2 Evaluation

We evaluate our optimization scheme with two metrics. First, we compare the log-likelihood on the test set obtained with the inferred model $\mathcal{L}$ to the test log-likelihood with the model that generated the data $\mathcal{L}^*$. We use this metric when the data is generated with a single set of parameters over the dataset (i.e., the same DPP matrix L is used to generate all the observations) as in such case the difference of test log-likelihood between two models ($\mathcal{L}^* - \mathcal{L}$) is an estimation of the Kullback-Leibler divergence between the two models.

We also consider a distance between the inferred embedding U and the embedding that generated the data U^*. As the performance is invariant to any permutation of column in the matrix U (together with indices of θ) and to a scaling factor — both (U, θ) and $(\frac{1}{\sqrt{\gamma}}U, \gamma\theta)$ correspond to the same DPP matrix L — we consider the following distance that compares the linear space produced with $U \in \mathbb{R}^{V \times r}$ and $U^* \in \mathbb{R}^{V \times r^*}$:

$$D(U, U^*) = \|U(U^\top U)^{-1}U^\top U^* - U^*\|_F / \|U^*\|_F,$$

where $\|.\|_F$ is the Frobenius norm. This distance is invariant to scaling and rotation and is equal to zero when U and U^* span the same space in $\mathbb{R}^V$. In particular, if we generate randomly the r columns of $Z \in \mathbb{R}^{V \times r}$, the expectation of the distance to U^* is $\mathbb{E}_Z[D(Z, U^*)] = 1 - \frac{r}{V}$. We display this quantity as "chance" in the following. As the number of columns in $U \in \mathbb{R}^{V \times r}$ and $U \in \mathbb{R}^{V \times r^*}$ is different, we can not use losses similar to what is used in independent component analysis [Hyvärinen et al., 2004]. The distance $D(U, U^*)$ seems appropriate here as it measures if we recover the correct subspace for a sufficiently small rank and allows us to compare matrices of different shapes.

CONTINUOUS SET $[0, 1]^2$. We compare our inference method to the best diagonal DPP $L_{\eta^*} = \eta^* I$, where $\eta^* \in \mathbb{R}$ maximizes the log-likelihood.

ITEMS SET, $\mathcal{X} = \{1, \dots, V\}$. We compare our inference method to the Picard iteration on full matrices proposed by Mariet and Sra [2015]. As they only consider the scenario where all the observations are drawn from the same DPP, we only compare to our method in that case.

4.5.3 Synthetic datasets

CONTINUOUS SET $[0, 1]^2$. We present the difference in log-likelihood between the inferred model and the model that generates the data as a function of the iterations in Figure 23. The comparison between the resulting kernel and the

kernel that generates the data is presented in Appendix 4.A. We observe that our model performs significantly better than the $\eta^* I$ kernel and converges to the the true log-likelihood.

ITEMS SET & EXPONENTIAL SET. We present the difference in log-likelihood and the distance of embeddings U between the inferred model and the model that generates the data as a function of the rank r of the representation in Figure 24 for the ground set $\mathcal{X} = \{1, \dots, V\}$ and in Figure 25 for the ground set $\mathcal{X} = \{0,1\}^V$. For set $\mathcal{X} = \{0,1\}^V$, results for observations generated from the same DPP (i.e., with a single θ for the whole dataset) are presented in Appendix 4.D as we recover the parameter U^* with the same precision for any regularization coefficient λ. We observe that the penalization may deteriorate the performance in terms of log-likelihood but significantly improves the quality of the recovered parameters. In practice, as our penalization $\mathcal{R}$ induces sparsity we recover sparse θ when $r > r^*$. For both ground sets, the parameter U^* that generated the data is recovered for $r^* < r < V$. In matrix factorization, increasing the size of the factors leads to fewer or no local minima [Haeffele et al., 2014], which is consistent with the fact that we only recover U^* for $r > r^*$.

For the items set $\mathcal{X} = \{1, \dots, V\}$, while the datasets are generated with $r^* = 5$, we observe the parameter U^* is only recovered when we optimize with $r \geq 30$. We also observe that our method performs better than the Picard iteration of Mariet and Sra [2015] in terms of log-likelihood. The Picard iteration updates the full matrix L and there is no tradeoff between the rank and the closeness of spanned subspaces, conveyed by $D(U, U^*)$.

For the exponential set $\mathcal{X} = \{0,1\}^V$, $r^* = 2$ and the parameter U^* is recovered for $r \geq 6$.

4.5.4 Real dataset.

Summaries with $l = 5$ sentences of two reviews are presented in Table 9. The corresponding embeddings U are presented in Table 10 and Table 11. We observe that our method is able to extract sentences that describes the opinion of the user on the restaurant. In particular, the sentences extracted with our method convey commitment of the user to aspects (food, service,...) while other sentences of the reviews only describe the context of the meal.

Columns of U

We present four embeddings (i.e., columns of $U \in \mathbb{R}^{V \times r}$) out of $r = 10$ learned on a restaurant reviews dataset with our DPP structure in Table 10 below. We display the 20 words with the highest absolute values for each column of U. We observe that our embeddings extract qualitative words (e.g., *good*, *great*, *friendly*). Even if the embeddings are not as consistent as topics extracted with topic models (e.g., LDA), we can distinguish different aspects of restaurants with the em-

beddings. For instance, words with positive values in embedding 1 are related to the food (e.g., *cream*, *love*, *crispy*, *tomato*); words with positive values in embedding 2 are associated to the service aspect (with *service*, *friendly*, *staff*, *attentive*). Moreover, they already lead to good summaries.

Table 10: Four embeddings (columns of U) inferred with $r = 10$ on restaurant reviews dataset.

Embed. 1	$U_{w,1}$	**Embed. 2**	$U_{w,2}$	**Embed. 3**	$U_{w,3}$	**Embed. 4**	$U_{w,4}$
love	0.19	service	0.74	great	0.99	place	0.75
could	0.11	friendly	0.36	food	0.8	great	0.41
large	0.11	nice	0.33	service	0.4	good	0.35
cream	0.1	good	0.24	star	0.32	really	0.28
crispy	0.1	pretty	0.24	worth	0.26	love	0.21
tomato	0.09	staff	0.24	place	0.26	nice	0.16
meat	0.09	price	0.15	price	0.25	service	0.16
ice	0.08	experience	0.14	back	0.21	atmosphere	0.14
sauce	0.08	well	0.14	wait	0.2	get	0.14
mouth	0.08	attentive	0.13	definitely	0.18	friendly	0.13
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
back	-0.48	would	-0.26	also	-0.27	could	-0.21
sushi	-0.51	think	-0.27	tasty	-0.28	dinner	-0.21
place	-0.51	try	-0.28	fresh	-0.31	menu	-0.25
pretty	-0.53	restaurant	-0.28	salad	-0.32	restaurant	-0.27
really	-0.58	one	-0.3	delicious	-0.36	well	-0.28
come	-0.64	amazing	-0.33	really	-0.4	come	-0.37
great	-0.73	like	-0.57	nice	-0.43	eat	-0.38
service	-0.79	get	-0.64	like	-0.44	food	-0.39
food	-1.08	love	-0.73	chicken	-0.45	time	-0.4
good	-2.08	place	-1.05	order	-0.59	price	-0.4

Rows of U

From the embeddings U, we can also compute similarity between words using the rows of U. We use the cosine similarity, i.e., for words $v, w \in \{1, \dots, V\}$:

$$\text{Cos}(v, w) = \frac{\langle U_v, U_w \rangle}{\|U_v\|_2 \|U_w\|_2},$$

where $U_v \in \mathbb{R}^r$ is the v^{th} row of U. We present ten examples of words with their closest words for cosine similarity in Table 11. We observe that our word embeddings also capture context from the sentences. For instance, the closest words to *food* are mostly adjective applicable to food (e.g., *solid*, *average*, *decent*, *expensive*). We observe the same characteristic for the words of the top row in Table 11. For adjectives of the bottom row in Table 11 (i.e., *good*, *tender*, *tasty* and *dry*), the closest words are either synonnyms/antonyms or nouns that may have the characteristic conveyed by the corresponding adjective. For instance, among the closest words to *tender*, the words *juicy* and *flavorful* have similar meaning than *tender*, *hard* is an antonym while *gnocchi*, *shrimp*, *sausage* may be characterized as *tender*. Finally, the closest words to *time* are mostly words that convey temporal meaning (e.g., *late*, *day*, *open*, *saturday*)

4.6 CONCLUSION

In this chapter, we propose a new class of determinantal point processes that can be run on a huge number of items because of a specific low-rank decomposition. This allowed parameter learning for continuous DPPs and new applications such as document modelling and summarization.

We apply our model on exponential set $\mathcal{X} = \{0, 1\}^V$ to model documents, it would be interesting to apply our inference to the infinite ground set $\mathcal{X} = \mathbb{N}^V$ as suggested in this chapter. We would also like to study the inference in continous exponential set $\mathcal{X} = \mathbb{R}^V$ using our decomposition.

While we focused primarily on DPPs to model diversity, it would also be interesting to consider other approaches based on submodularity [Djolonga and Krause, 2014, Djolonga et al., 2016] and study the tractability of these models for exponantially large numbers of items.

We acknowledge support from the CIFAR program in Learning in Machines & Brains.

Table 11: Ten examples of cosine similarity between words (i.e., between rows of U) with $r = 10$ on restaurant reviews dataset.

food	Cos
solid	0.97
delivery	0.91
average	0.9
indian	0.9
decent	0.88
overall	0.86
expensive	0.85
quality	0.83
italian	0.83
sunday	0.82

service	Cos
slow	0.93
friendly	0.91
fast	0.9
quick	0.88
delivery	0.87
extremely	0.85
staff	0.83
experience	0.8
average	0.77
good	0.75

decor	Cos
unique	1.0
vibe	0.95
warm	0.87
date	0.87
atmosphere	0.85
damn	0.82
cool	0.81
beach	0.79
broth	0.76
run	0.73

atmosphere	Cos
cool	0.94
unique	0.88
view	0.85
wonderful	0.85
decor	0.85
fun	0.82
vibe	0.82
date	0.81
kind	0.79
pancake	0.78

meal	Cos
cheap	0.97
drink	0.97
sunday	0.96
though	0.96
sushi	0.94
city	0.93
overall	0.92
visit	0.91
well	0.91
bad	0.91

good	Cos
location	0.98
look	0.96
hit	0.93
bad	0.9
ever	0.9
quick	0.87
okay	0.87
pretty	0.86
sure	0.85
city	0.85

tender	Cos
juicy	0.96
hard	0.95
flavorful	0.93
light	0.93
gnocchi	0.89
shrimp	0.89
sausage	0.89
real	0.88
water	0.88
main	0.87

tasty	Cos
awesome	0.99
fresh	0.97
delicious	0.96
people	0.95
course	0.92
beer	0.92
fill	0.92
fish	0.91
nice	0.9
server	0.9

dry	Cos
light	0.93
inside	0.93
ingredient	0.92
salty	0.91
potato	0.9
sausage	0.89
meat	0.89
put	0.88
tender	0.85
kinda	0.85

time	Cos
late	0.97
day	0.97
open	0.97
first	0.96
saturday	0.93
far	0.92
visit	0.91
last	0.9
though	0.89
price	0.88

APPENDIX

4.A CONTINUOUS SET $[0,1]^2$

In this section, we present a comparison between the true marginal kernel (that generates the data) K^* and the inferred marginal kernel K_t. More precisely, $\mathcal{X} = [0,1]^2$ and we compute the induced distance from the center point $q = (\frac{1}{2}, \frac{1}{2})$ to any point $x \in \mathcal{X}$, i.e., $K(x,q)$. We show in Figure 26 a comparison between the true distance $K^*(x,q)$ and the inferred distance $K_t(x,q)$ after $t = 100$ iterations.

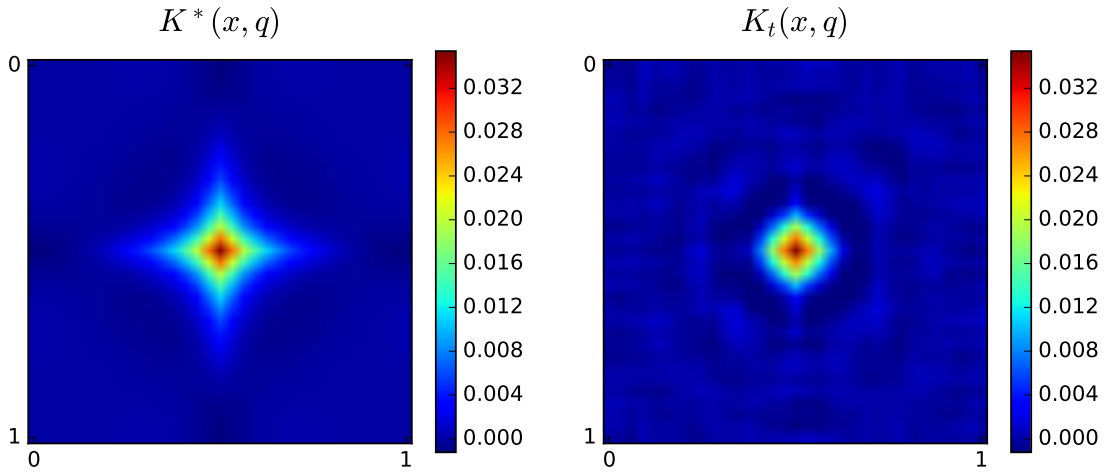


Figure 26: Comparison of K^* and K_t.

4.B PICARD ITERATION

We apply the Picard iteration of Mariet and Sra [2015] on the synthetic "items" datasets (i.e., observations are generated from $L = \alpha I + U \operatorname{Diag}(\theta) U^\top$) with $N = 100$ items. We present the evolution of the objective function through the iterations with the Picard iteration in Figure 27. We observe a similar evolution than presented in the original paper [Mariet and Sra, 2015]. This however led in Figure 24 to a lower likelihood than L-BFGS on U.

4.C EXPONENTIAL SET $\mathcal{X} = \{0,1\}^V$

We present the difference in log-likelihood and the distance if embeddings U between the inferred model and the model that generates data as a function of the rank r of the representation in Figure 28.

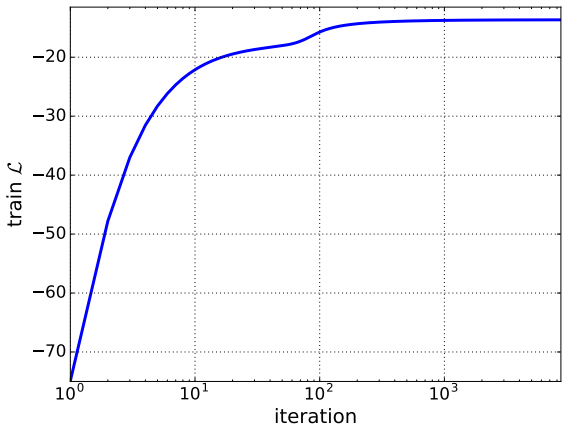
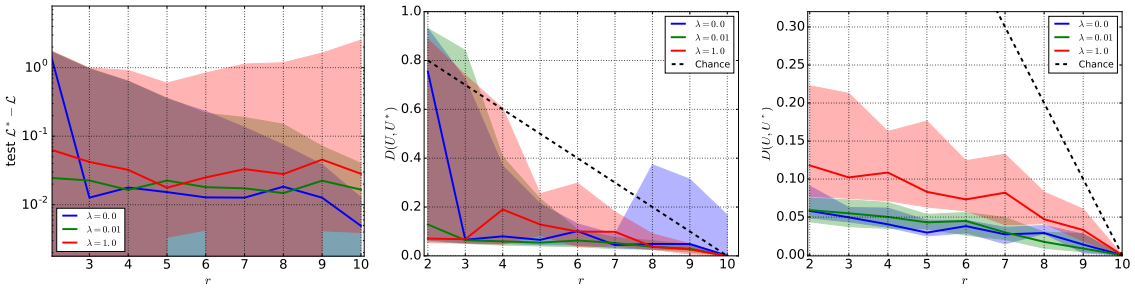


Figure 27: Picard iteration [Mariet and Sra, 2015]. Evolution of the objective function (train log-likelihood) as a function of the iterations.



(a) Distance in log-likelihood. (b) Distance between U and U^*. (c) Distance between U and U^*.

Figure 28: Performance for ground set $\mathcal{X} = \{0, 1\}^V$ as a function of r. (a,b) Same θ for all the observations; (c) A different θ for each observation.

4.D SUMMARY AS A SUBSAMPLE – PARAMETER LEARNING

We assume that $\sum_{x \in \mathcal{X}} p(x)\phi(x)\phi(x)^\top = \operatorname{Diag}(v) + \mu\mu^\top$. The log-likelihood of an observed document X is expressed as $\ell(X|L) = \log\det L_X - \log\det(L + I)$. The computation of the second term, $\log\det(L + I)$, is untractable to compute in reasonable time for any L when $V \geq 20$, since $L \in \mathbb{R}^{2^V \times 2^V}$. We can still compute this value for structured L. When $L = \alpha I + \operatorname{Diag}(p)^{1/2}\Phi A \Phi^\top \operatorname{Diag}(p)^{1/2}$, we have, using the matrix determinant lemma and Woodbury identity:

$$\det(L + I) = \det[(\alpha + 1)I] \det A \det\left(A^{-1} + \frac{1}{\alpha + 1}\Phi^\top \operatorname{Diag}(p)\Phi\right).$$

We then have, if $\rho = \frac{1}{\alpha+1}$:

$$\begin{aligned}\log\det\left(A^{-1} + \rho\,\mathrm{Diag}(\nu) + \rho\mu\mu^\top\right) &= \log\left[1 + \rho\mu^\top\left(A^{-1} + \rho\,\mathrm{Diag}(\nu)\right)^{-1}\mu\right] \\ &\quad + \log\det(A^{-1} + \rho\,\mathrm{Diag}(\nu)) \\ &\quad \text{(matrix determinant lemma)}\end{aligned}$$

$$\begin{aligned}= \log\left[1 + \mu^\top\left(\mathrm{Diag}(1/\nu) - \mathrm{Diag}(1/\nu)(\rho A + \mathrm{Diag}(1/\nu))^{-1}\mathrm{Diag}(1/\nu)\right)\mu\right] \\ + \log\det(A^{-1} + \rho\,\mathrm{Diag}(\nu)) \\ \text{(Woodbury identity).}\end{aligned}$$

If we consider $A = \gamma I + U\,\mathrm{Diag}(\theta)U^\top$, we have:

$$\begin{aligned}(\rho A + \mathrm{Diag}(1/\nu))^{-1} &= \left[(\rho\gamma I + \rho U\,\mathrm{Diag}(\theta)U^\top + \mathrm{Diag}(1/\nu)\right]^{-1} \\ &= \mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right) \\ &\quad - \mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right)U \\ &\quad \times \left(\mathrm{Diag}(1/\rho\theta) + U^\top\,\mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right)U\right)^{-1} \\ &\quad \times U^T\,\mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right),\end{aligned}$$

$$\begin{aligned}&\log\det(A^{-1} + \rho\,\mathrm{Diag}(\nu)) \\ &\quad = \log\det\left[\mathrm{Diag}(\frac{1}{\gamma} + \rho\nu) - \frac{1}{\gamma}U\left(\mathrm{Diag}(\gamma/\theta) + U^\top U\right)^{-1}U^\top\right] \\ &\quad\quad \text{(Woodbury identity on } A^{-1}) \\ &\quad = \log\det\left[\left(\mathrm{Diag}(\gamma/\theta) + U^\top U\right) - \frac{1}{\gamma}U^\top\,\mathrm{Diag}\left(\frac{\gamma}{1+\nu\gamma\rho}\right)U\right] \\ &\quad\quad - \log\det(\mathrm{Diag}(\gamma/\theta) + U^\top U) + \log\det(\frac{I}{\gamma} + \rho\,\mathrm{Diag}(\nu)) \\ &\quad = \log\det\left[\mathrm{Diag}(\gamma/\theta) + U^\top\,\mathrm{Diag}\left(\frac{\nu\gamma\rho}{1+\nu\gamma\rho}\right)U\right] \\ &\quad\quad - \log\det(\mathrm{Diag}(\gamma/\theta) + U^\top U) + \log\det(\frac{I}{\gamma} + \rho\,\mathrm{Diag}(\nu)),\end{aligned}$$

$$\log\det(A) = \log\det(\mathrm{Diag}(1/\theta) + \frac{1}{\gamma}U^\top U) + \sum_k \log\theta_k + V\log\gamma.$$

In the end, the computation of $\log\det(L+I)$ only needs matrix products of size V and inversions of size r.

5 CONCLUSION

The ultimate goal of this work is to suggest any user a personalized list of contents with a short readable text attached to each content, where this short text conveys the opinion the user may form about the corresponding content. We proposed a line of work towards this goal. In particular, we present new topic models and new inference schemes for these models to help the users quickly assess previously unseen contents. Given the work developped in this thesis, the possible future directions are the following:

- Througout this thesis, we evaluate the performance of the presented methods with predictive likelihood (e.g., following Wallach et al. [2009] for LDA). In particular, we only use empirical results—such as topics extracted with our methods—to illustrate the models when applied to real datasets. Following a heavy line of work on the evaluation of topic coherence such as Newman et al. [2010], Mimno et al. [2011], Lau et al. [2014], Röder et al. [2015], it would be interesting to measure the impact of our methods on topic coherence. It would also be interesting to include the presented models in a recommender systems and measure the impact of the model on the speed of choice and the satisfaction of the user.

- We could study topic models in decentralized networks [Colin and Dupuy, 2016], i.e., networks with limited communication between nodes. In such networks, it is typically impossible to efficiently centralize data or to globally aggregate intermediate results: agents can only communicate with their immediate neighbors, often in a completely asynchronous fashion. It would be interesting to build an inference scheme for topic models suited to such networks and investigate empirical differences between topics extracted in a decentralized settings compared to online inference proposed in Chapter 3.

- Following Brunel et al. [2017] and Urschel et al. [2017], it would be interesting to investigate the theoretical convergence guarantees for the maximum likelihood estimator of DPP as presented in Chapter 4. More specifically, as we propose a specific low-rank factorization of the marginal kernel, we would like to look for guarantees on the maximum likelihood estimator for such low-rank matrices.

- Moreover, we apply the DPP model presented in Chapter 4 to text summarization, where we are able to summarize a single review with readable sentences. It would be interesting to build a model able to summarize a

full corpus of documents. This kind of model would be for instance useful in crowd-sourced review services (such as Yelp or IMDB) where we could automatically suggest the user the most relevant sentences among all the previous reviews. In particular, it would be interesting to enhance the DPP model with other information than text only (e.g., the ratings, the *useful* score, content class such that genre for movies of type of food for restaurants) in order to personalize the recommendation and get closer to the ultimate goal presented at the beginning of this thesis. While we are able to extract word embeddings with our formulation of DPPs, it would be interesting to extract other embeddings, for instance on the genres of movies or the types of products, in order to compute distances between these genres or types of products.

- We also would like to apply the DPP model of Chapter 4 to computer vision. More specifically, we would like to tackle the problem of diversity of the outputs proposed by computer vision models. For instance, let us consider the problem of multi-class segmentation. The goal is to accurately assign a single class to every pixel among K possible classes. While existing models cast this problem as an optimization problem, there has recently been an interest in extracting diverse quality outputs from the model [Kirillov et al., 2015, Batra et al., 2012, Dey et al., 2015]. In this problem of mutli-class segmentation, if you select the M segmentations with the best values for the objective function, it is very likely that these M outputs only differ from only few pixels. The motivations for diverse outputs resides in the fact that computer vision tasks are often addressed by a series of modules (e.g., layers in a neural network) where each module generates several hypotheses as input to the next module. A good practice in this structure is to consider diverse options at each module in order to avoid premature commitment to a low quality feature that would destroy the quality of the final output [Viola and Jones, 2001, Felzenszwalb and McAllester, 2007]. In this context, the DPPs are perfectly suited as a model to generate diverse outputs. In the particular case of the multi-class segmentation problem, it would be interesting to learn a distance between the K possible classes (through class embeddings) from examples of such segmentations and to have the possibility to sample diverse segmentations. Another computer vision problem is the estimation of the pose of persons in an image where it is very unlikely that two persons stands at the same place in a picture. In this case, the DPP could promote spaced pose estimations with a higher probability.

6

REFERENCES

R. Affandi, E. Fox, and B. Taskar. Approximate inference in continuous determinantal processes. In *Adv. NIPS*, 2013.

R. Affandi, E. Fox, R. Adams, and B. Taskar. Learning the parameters of determinantal point process kernels. In *Proc. ICML*, 2014.

C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012.

L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi. Topic significance ranking of LDA generative models. In *Proc. ECML*, 2009.

K. Atkinson and W. Han. *Spherical Harmonics and Approximations on the Unit Sphere: an Introduction*, volume 2044. Springer, 2012.

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Adv. NIPS*, 2013.

R. Bardenet and M. Titsias. Inference for determinantal point processes without spectral knowledge. In *Adv. NIPS*, 2015.

D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse m-best solutions in Markov random fields. In *Proc. ECCV*, 2012.

J. Becker and D. Kuropka. Topic-based vector space model. In *Proc. ICBIS*, 2003.

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. Blei and J. Lafferty. Dynamic topic models. In *Proc. ICML*, 2006.

D. Blei and J. Lafferty. A correlated topic model of Science. *The Annals of Applied Statistics*, pages 17–35, 2007.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

D. Blei, T. Griffiths, and M. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.

A. Borodin and E. Rains. Eynard–Mehta theorem, Schur process, and their Pfaffian analogs. *Journal of statistical physics*, 121(3):291–317, 2005.

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998.

T. Broderick, N. Boyd, A. Wibisono, A. Wilson, and M. Jordan. Streaming variational Bayes. 2013.

P. Brown, P. Desouza, R. Mercer, V. Pietra, and J. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

V.-E. Brunel, A. Moitra, P. Rigollet, and J. Urschel. Maximum likelihood estimation of determinantal point processes. *arXiv preprint arXiv:1701.06501*, 2017.

W. Buntine and A. Jakulin. Discrete principal component analysis. In *Proc. of the Subspace, Latent Structure and Feature Selection Techniques: Statistical and Optimisation perspectives Workshop*, 2005.

O. Cappé and E. Moulines. Online EM algorithm for latent data models. *Journal of the Royal Statistical Society*, 71(3):593–613, 2009.

G. Casella and R. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

G. Casella and E. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei. Reading tea leaves: How humans interpret topic models. 2009.

I. Colin and C. Dupuy. Decentralized topic modelling with latent Dirichlet allocation. *arXiv preprint arXiv:1610.01417*, 2016.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 1990.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, 1999.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39(1):1–38, 1977.

D. Dey, V. Ramakrishna, M. Hebert, and J. Bagnell. Predicting multiple structured visual interpretations. In *Proc. IEEE ICCV*, pages 2947–2955, 2015.

I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *Adv. NIPS*, 2005.

Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proc. ACM SIGKDD*, 2014.

J. Djolonga and A. Krause. From MAP to marginals: Variational inference in Bayesian submodular models. In *Adv. NIPS*, 2014.

J. Djolonga, S. Tschiatschek, and A. Krause. Variational inference in mixed probabilistic submodular models. In *Adv. NIPS*, 2016.

S. Dumais. Latent semantic indexing (LSI). In *The Second Text REtrieval Conference (TREC-2)*, 1994.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

P. Felzenszwalb and D. McAllester. The generalized A* architecture. *Journal of Artificial Intelligence Research*, 29:153–190, 2007.

J. Foulds, S. Kumar, and L. Getoor. Latent topic networks: A versatile probabilistic programming framework for topic models. In *Proc. ICML*, 2015.

Y. Gao, J. Chen, and J. Zhu. Streaming Gibbs sampling for LDA model. *arXiv preprint arXiv:1601.01142*, 2016.

M. Gartrell, U. Paquet, and N. Koenigstein. Low-rank factorization of determinantal point processes for recommendation. *arXiv preprint arXiv:1602.05436*, 2016.

J. Gillenwater, A. Kulesza, and B. Taskar. Discovering diverse and salient threads in document collections. In *Proc. EMNLP*, 2012a.

J. Gillenwater, A. Kulesza, and B. Taskar. Near-optimal MAP inference for determinantal point processes. In *Adv. NIPS*, 2012b.

J. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-maximization for learning determinantal point processes. In *Adv. NIPS*, 2014.

T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proc. CogSci*, 2002.

T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

B. Haeffele, E. Young, and R. Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proc. ICML*, 2014.

Z. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

D. Hiemstra. A probabilistic justification for using tf×idf term weighting in information retrieval. *International Journal on Digital Libraries*, 3(2):131–139, 2000.

M. Hoffman and D. Blei. Structured stochastic variational inference. In *Proc. AISTATS*, 2015.

M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. 2010.

M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

T. Hofmann. Probabilistic latent semantic analysis. In *Proc. UAI*, 1999a.

T. Hofmann. Probabilistic latent semantic indexing. In *Proc. ACM SIGIR*, 1999b.

G. Huang, C. Guo, M. Kusner, Y. Sun, F. Sha, and K. Weinberger. Supervised word mover's distance. In *Adv. NIPS*, 2016.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

F. Jelinek and R. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice*, 1980.

Y. Jo and A. Oh. Aspect and sentiment unification model for online review analysis. In *Proc. ACM WSDM*, 2011.

B. Kang. Fast determinantal point process sampling with application to clustering. In *Adv. NIPS*, 2013.

N. Kantas, A. Doucet, S. Singh, J. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30 (3):328–351, 2015.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.

A. Kirillov, B. Savchynskyy, D. Schlesinger, D. Vetrov, and C. Rother. Inferring m-best diverse labelings in a single one. In *Proc. IEEE ICCV*, 2015.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. ICASSP*, 1995.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012.

A. Kulesza and B. Taskar. k-DPPs: Fixed-size determinantal point processes. In *Proc. ICML*, 2011.

A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

H. Kushner and G. Yin. *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, second edition, 2003.

M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *Proc. ICML*, 2015.

J. Han Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, 2014.

F. Lavancier, J. Møller, and E. Rubak. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4):853–877, 2015.

D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

E. Lehmann and G. Casella. *Theory of point estimation*, volume 31. Springer Science & Business Media, 1998.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

A. Lewis and M. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, 2013.

C. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes. In *Proc. AISTATS*, 2016a.

C. Li, S. Jegelka, and S. Sra. Fast DPP sampling for Nystrom with application to kernel methods. In *Proc. ICML*, 2016b.

P. Liang and D. Klein. Online EM for unsupervised models. In *Proc. NAACL HLT*, 2009.

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proc. ACM CIKM*, 2009.

G. Ling, M. R. Lyu, and I. King. Ratings meet reviews, a combined approach to recommend. In *Proc. ACM RecSys*, 2014.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv preprint arXiv:1402.4419*, 2014.

Z. Mariet and S. Sra. Fixed-point algorithms for learning determinantal point processes. In *Proc. ICML*, 2015.

Z. Mariet and S. Sra. Kronecker determinantal point processes. *arXiv preprint arXiv:1605.08374*, 2016.

J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proc. ACM RecSys*, 2013.

J. Mcauliffe and D. Blei. Supervised topic models. In *Adv. NIPS*, 2008.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proc. ACM WWW*, 2007.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Adv. NIPS*, 2013b.

D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proc. EMNLP*, 2011.

D. Mimno, M. Hoffman, and D. Blei. Sparse stochastic inference for latent Dirichlet allocation. *Proc. ICML*, 2012.

T. Minka. Estimating a Dirichlet distribution. Technical report, 2000.

C. Mooers. The theory of digital handling of non-numerical information and its implications to machine economics. *Proc. ACM at Rutgers University*, 1950.

K. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

D. Newman, J. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *NAACL HLT*, 2010.

D. Newman, E. Bonilla, and W. Buntine. Improving topic coherence with regularized topic models. *Adv. NIPS*, 2011.

K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, 1999.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2-3):103–134, 2000.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

J. Paisley, C. Wang, D. Blei, and M. Jordan. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

K. Palla, F. Caron, and Y. Teh. Bayesian nonparametrics for sparse dynamic networks. *arXiv preprint arXiv:1607.01624*, 2016.

S. Patterson and Y. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Adv. NIPS*, 2013.

M. Paul and M. Dredze. Factorial LDA: Sparse multi-dimensional text models. In *Adv. NIPS*, 2012.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *Proc. ACL*, 1993.

A. Podosinnikova, F. Bach, and S. Lacoste-Julien. Rethinking LDA: moment matching for discrete ICA. In *Adv. NIPS*, 2015.

B. Polyak and A. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

J. Rennie. *Improving multi-class text classification with naive Bayes*. PhD thesis, MIT, 2001.

F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proc. ACM WSDM*, 2015.

D. Rohde and O. Cappé. Online maximum-likelihood estimation for latent factor models. In *Proc. IEEE SSP Workshop*, 2011.

G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

I. Sato, K. Kurihara, and H. Nakagawa. Deterministic single-pass algorithm for LDA. In *Adv. NIPS*, 2010.

B. Scholkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proc. ACM WWW*, 2008.

M. Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):257–267, 1984.

J. Urschel, V.-E. Brunel, A. Moitra, and P. Rigollet. Learning determinantal point processes with moments and cycles. *arXiv preprint arXiv:1703.00539*, 2017.

A. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE CVPR*, 2001.

H. Wallach. Topic modeling: beyond bag-of-words. 2006.

H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proc. ICML*, 2009.

C. Wang and D. Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *Adv. NIPS*, 2012.

C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

F. Yan, N. Xu, and Y. Qi. Parallel inference for latent Dirichlet allocation on graphics processing units. In *Adv. NIPS*, 2009.

H. Zhao, B. Jiang, and J. Canny. SAME but different: Fast and high-quality Gibbs parameter estimation. *arXiv preprint arXiv:1409.5402*, 2014.

W. Zou, R. Socher, D. Cer, and C. Manning. Bilingual word embeddings for phrase-based machine translation. In *Proc. EMNLP*, 2013.

Résumé

La plupart des systèmes de recommandation actuels se base sur des évaluations sous forme de notes (i.e., chiffre entre 0 et 5) pour conseiller un contenu (film, restaurant...) à un utilisateur. Ce dernier a souvent la possibilité de commenter ce contenu sous forme de texte en plus de l'évaluer. Il est difficile d'extraire de l'information d'un texte brut tandis qu'une simple note contient peu d'information sur le contenu et l'utilisateur. Dans cette thèse, nous tentons de suggérer à l'utilisateur un texte lisible personnalisé pour l'aider à se faire rapidement une opinion à propos d'un contenu.

Plus spécifiquement, nous construisons d'abord un modèle thématique prédisant une description de film personnalisée à partir de commentaires textuels. Notre modèle sépare les thèmes qualitatifs (i.e., véhiculant une opinion) des thèmes descriptifs en combinant des commentaires textuels et des notes sous forme de nombres dans un modèle probabiliste joint. Nous évaluons notre modèle sur une base de données IMDB et illustrons ses performances à travers la comparaison de thèmes.

Nous étudions ensuite l'inférence de paramètres dans des modèles à variables latentes à grande échelle, incluant la plupart des modèles thématiques. Nous proposons un traitement unifié de l'inférence en ligne pour les modèles à variables latentes à partir de familles exponentielles non-canoniques et faisons explicitement apparaître les liens existants entre plusieurs méthodes fréquentistes et Bayesiennes proposées auparavant. Nous proposons aussi une nouvelle méthode d'inférence pour l'estimation fréquentiste des paramètres qui adapte les méthodes MCMC à l'inférence en ligne des modèles à variables latentes en utilisant proprement un échantillonnage de Gibbs local. Pour le modèle thématique d'allocation de Dirichlet latente, nous fournissons une vaste série d'expériences et de comparaisons avec des travaux existants dans laquelle notre nouvelle approche est plus performante que les méthodes proposées auparavant.

Enfin, nous proposons une nouvelle classe de processus ponctuels déterminantaux (PPD) qui peut être manipulée pour l'inférence et l'apprentissage de paramètres en un temps potentiellement sous-linéaire en le nombre d'objets. Cette classe, basée sur une factorisation spécifique de faible rang du noyau marginal, est particulièrement adaptée à une sous-classe de PPD continus et de PPD définis sur un nombre exponentiel d'objets. Nous appliquons cette classe à la modélisation de documents textuels comme échantillons d'un PPD sur les phrases et proposons une formulation du maximum de vraisemblance conditionnel pour modéliser les proportions de thèmes, ce qui est rendu possible sans aucune approximation avec notre classe de PPD. Nous présentons une application à la synthèse de documents avec un PPD sur 2^{500} objets, où les résumés sont composés de phrases lisibles.

Mots Clés

modèles thèmatiques, apprentissage en ligne, modèles à variables latentes, apprentissage non supervisé, processus ponctuels determinantaux, allocation de Dirichlet latente.

Abstract

Most of current recommendation systems are based on ratings (i.e. numbers between 0 and 5) and try to suggest a content (movie, restaurant...) to a user. These systems usually allow users to provide a text review for this content in addition to ratings. It is hard to extract useful information from raw text while a rating does not contain much information on the content and the user. In this thesis, we tackle the problem of suggesting personalized readable text to users to help them make a quick decision about a content.

More specifically, we first build a topic model that predicts personalized movie description from text reviews. Our model extracts distinct qualitative (i.e., which convey opinion) and descriptive topics by combining text reviews and movie ratings in a joint probabilistic model. We evaluate our model on an IMDB dataset and illustrate its performance through comparison of topics.

We then study parameter inference in large-scale latent variable models, which include most topic models. We propose a unified treatment of online inference for latent variable models from a non-canonical exponential family, and draw explicit links between several previously proposed frequentist or Bayesian methods. We also propose a novel inference method for the frequentist estimation of parameters, which adapts MCMC methods to online inference of latent variable models with the proper use of local Gibbs sampling. For the specific latent Dirichlet allocation topic model, we provide an extensive set of experiments and comparisons with existing work, where our new approach outperforms all previously proposed methods.

Finally, we propose a new class of determinantal point processes (DPPs) which can be manipulated for inference and parameter learning in potentially sublinear time in the number of items. This class, based on a specific low-rank factorization of the marginal kernel, is particularly suited to a subclass of continuous DPPs and DPPs defined on exponentially many items. We apply this new class to modelling text documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions, which is made possible with no approximation for our class of DPPs. We present an application to document summarization with a DPP on 2^{500} items, where the summaries are composed of readable sentences.

Keywords

topic models, online learning, latent variable models, unsupervised learning, determinantal point processes, latent Dirichlet allocation.