

Streaming-LDA: A Copula-based Approach to Modeling Topic Dependencies in Document Streams

Hesam Amoualian, Marianne Clausel, Eric Gaussier, Massih-Reza Amini

▶ **To cite this version:**

Hesam Amoualian, Marianne Clausel, Eric Gaussier, Massih-Reza Amini. Streaming-LDA: A Copula-based Approach to Modeling Topic Dependencies in Document Streams. 22nd ACM SIGKDD Conference Knowledge Discovery and Data Mining, Aug 2016, San Francisco, United States. pp.695-704 10.1145/2939672.2939781 . hal-01344779

HAL Id: hal-01344779
https://hal.science/hal-01344779v1

Submitted on 12 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Streaming-LDA: A Copula-based Approach to Modeling Topic Dependencies in Document Streams

Hesam Amoualian
University of Grenoble Alps - CNRS / LIG
hesam.amoualian@imag.fr

Marianne Clausel
University of Grenoble Alps - CNRS / LJK
marianne.clausel@imag.fr

Eric Gaussier
University of Grenoble Alps - CNRS / LIG
eric.gaussier@imag.fr

Massih-Reza Amini
University of Grenoble Alps - CNRS / LIG
massih-reza.amini@imag.fr

ABSTRACT

We propose in this paper two new models for modeling topic and word-topic dependencies between consecutive documents in document streams. The first model is a direct extension of Latent Dirichlet Allocation model (LDA) and makes use of a Dirichlet distribution to balance the influence of the LDA prior parameters wrt to topic and word-topic distribution of the previous document. The second extension makes use of copulas, which constitute a generic tools to model dependencies between random variables. We rely here on Archimedean copulas, and more precisely on Franck copulas, as they are symmetric and associative and are thus appropriate for exchangeable random variables. Our experiments, conducted on three standard collections that have been used in several studies on topic modeling, show that our proposals outperform previous ones (as dynamic topic models and temporal LDA), both in terms of perplexity and for tracking similar topics in a document stream.

CCS Concepts

•**Computing methodologies** → *Latent Dirichlet allocation;* •**Mathematics of computing** → *Bayesian computation; Gibbs sampling; Metropolis-Hastings algorithm;*

Keywords

Latent Dirichlet allocation, Copulas, Document Streams, Topic Dependencies

1. INTRODUCTION

The recent proliferation of temporal textual data on the Internet such as Tweets or comments on Youtube has brought new challenges for learning with interdependent data. Though important progress has been made in some directions [8], popular approaches for most of these tasks are designed to deal with static collections of documents. This is specially the case for latent topic modeling, albeit analyzes of social content have gained much attention in recent years for different aspects of daily life, such as latent health-related topic analysis [19] or buzz detection [20].

Although the main goal of probabilistic modeling is to find word topics, an equally interesting objective is to examine topic evolutions and transitions. The seminal work of [4] proposed to model the dynamic evolution of topics by first grouping documents into time slices and then to chain the evolution of both the word-topic and topic mixture distributions via a Gaussian process. In some cases, the Gaussian distribution was not found to be the appropriate distribution in modeling the topic shifts and some studies considered other probability distributions for capturing the evolution of topics over time [22]. However, the idea of grouping documents into epochs for modeling topic evolution was echoed in a number of studies. For example, [24] estimated a transition matrix over topic vectors between two predefined epochs and they showed that the LDA model [5] can be enhanced by considering directly the evolution of the topics over time.

In this paper we propose two extensions of LDA for modeling the dependency between two consecutive documents in a stream. In our first model, we suppose that the dependency between topic distributions of two consecutive documents follows a Dirichlet distribution controlled by an hyperparameter. This model is similar to the one of [4] with time slices equal to 1, but it offers a more precise mechanism for controlling the dependencies and is based on a framework encompassing all the situations (from complete independence to plain equality). This first study paves the way for a more general topic model in which the dependencies between the topics of two consecutive documents are captured by copulas which constitute generic tools to model dependencies between random variables [6]. Among the several families of copulas that have been defined in the literature, our choice fell on Archimedean copulas [13, 14] as they are symmetric and associative, necessary conditions when dealing with exchangeable random variables [18]. More particularly, we use Franck copulas, a special case of Archimedean copulas that rely on a single parameter, easier to estimate and more robust to sparse data. Using three collections with different characteristics, we show that our approaches are faster and improve over state-of-the-art topic models. We also analyze

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

© 2016 ACM. ISBN 978-1-4503-4232-2/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2939672.2939781

the precision of our models to track the topics on a labeled dataset.

The outline of this paper is as follows. In the next section, we present our models. In Section 3, we introduce an efficient procedure to estimate the most important, in terms of size, parameters. We then describe in Section 4 the experimental results obtained with our approaches on three distinct datasets. In Section 5, we position our work with respect to the state of the art. Finally, Section 6 concludes our study by summarizing its main results and by giving some pointers to future research.

2. STREAMING LDA

Latent Dirichlet Allocation (LDA, [5]) is a probabilistic Bayesian model used to describe a corpus of D documents, associated with a vocabulary of size V. In this model, latent variables, indexed in $\{1, \cdots, K\}$, are used to represent the *hidden* (in the sense non-observed) topics underlying each document. LDA is associated to the following generative model[1]:

- Generate, for each topic $k, 1 \leq k \leq K$, a distribution over the words: $\phi_k \sim Dir(\beta)$, where ϕ_k and β are V dimensional vectors;
- For each document d:
  - Choose a distribution over the topics: $\theta^d \sim Dir(\alpha)$, where θ^d and α are K dimensional vectors;
  - For each position (indexed by n, $1 \leq n \leq N$) in d: (a) Choose a topic assignment: $z_n^d \sim mult(1, \theta^d)$; (b) Choose the word w_n^d from the topic z_n^d with probability $P(w_n^d = v | z_n^d = k) = \phi_{k,v}$;

where N is the length of each document and $\phi_{k,v}$ is the v^{th} coordinate of ϕ_k. α and β correspond to the priors of the model. They are usually fixed, following [5]. Furthermore, in almost all previous studies on LDA, the priors are considered to be symmetric, each coordinate of the vector being equal: $\alpha_1 = \cdots = \alpha_K$. If one assumes a broad Gamma prior for both α and β, then their value can be easily learned from data by *maximum a posteriori* [1] or *Markov Chain Monte Carlo* [15] methods. One can also envisage learning asymmetric Dirichlet priors [21], which raises no particular difficulties for the models we are considering. For clarity sake, we however assume here fixed, symmetric priors; the extension to their learning through Gamma priors or through asymmetric priors is purely technical. In the remainder, we will denote by α and β the priors for the Dirichlet distributions as well the constant value taken by each coordinate of these priors, the context being sufficient to determine which element is referred to.

An important characteristic of LDA is that each document is generated independently from the previous ones. This is not a realistic assumption in different settings, as document streams, and we introduce below two extensions of LDA that model such dependencies.

2.1 Dirichlet-based dependencies

We introduce here a first extension of LDA, that we refer to as ST-LDA-D.

[1]For simplification and following standard practice, we do not model here the length of each document, assumed to be fixed and equal to N.

2.1.1 Presentation of the model

In this first model, we rely on a direct extension of the LDA model to take into account dependencies between the document-specific topic distributions of two sequential documents, denoted $(d-1)$ and d $(2 \leq d \leq D)$. This extension uses, as the standard LDA model, Dirichlet distributions for the document-specific topic distributions, the parameters of which are linear combination of the standard prior α and the topic distribution estimated in the previous document:

$$\theta^d | \theta^{d-1} \sim Dir(\alpha + \lambda_d \theta^{d-1}) \tag{1}$$

where λ_d is a uniformly distributed parameter that controls the influence of the topics of document $(d-1)$ on the topics of document d (see Figure 1). The expectation of each component of θ^d is given by:

$$\mathbb{E}[\theta_i^d | \theta_i^{d-1}] = \frac{\alpha + \lambda_d \theta_i^{d-1}}{K\alpha + \lambda_d} \tag{2}$$

Hence, if λ_d is high, i.e. if document d covers the same topics as document $(d-1)$, then $\mathbb{E}[\theta_i^d | \theta_i^{d-1}] \approx \theta_i^{d-1}$.

We furthermore assume that the previous document, $(d-1)$, can influence the word-topic distributions of the current document d. This assumption, also made in dynamic topic models [4] and topic tracking models [11], is motivated by the fact that, within a given topic, if word distributions evolve over time, they tend to do so in a smooth way. As before, one can use a direct extension of the LDA model to account for dependencies between word-topic distributions in sequential documents:

$$\forall k, 1 \leq k \leq K, \phi_k^d | \phi_k^{d-1} \sim Dir(\beta + \mu_d \phi_k^{d-1}) \tag{3}$$

Here μ_d is again a uniformly distributed parameter that controls the tradeoff between the prior β and the learned topic-word distributions ϕ^{d-1}. As usual ϕ_k^{d-1} is the word distribution of topic k. The conditional mean of each component of ϕ_k^d is given by:

$$\mathbb{E}[\phi_k^d | \phi^{d-1}] = \frac{\beta + \mu_d \phi_k^{d-1}}{V\beta + \mu_d} \tag{4}$$

and is approximately the value of the same component of document $(d-1)$ when the two documents are strongly dependent.

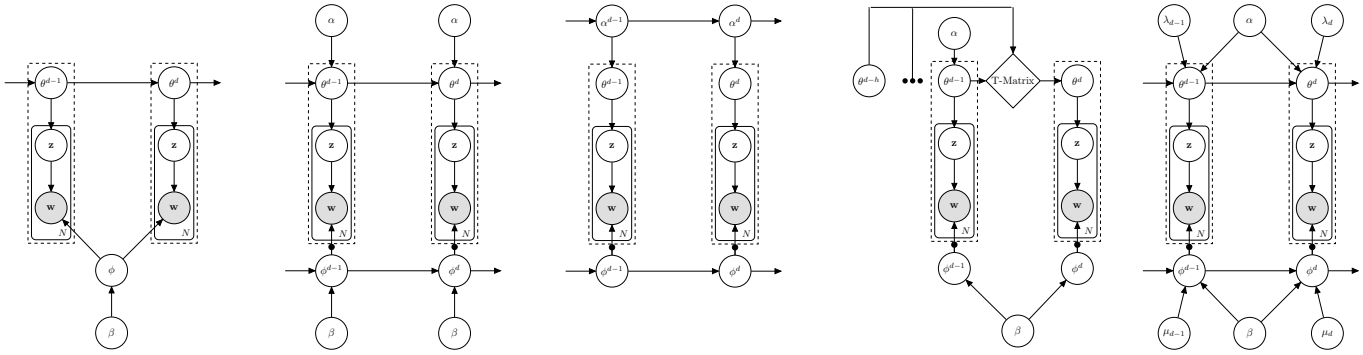
Lastly, as one can note, by setting $\lambda_d = \mu_d = 0$, $\forall d, 2 \leq d \leq D$, one "forgets" the dependencies between consecutive documents. The streaming model is in this case identical to the standard LDA model.

2.1.2 Inference with Gibbs sampling

As mentioned before, the parameters α and β are considered fixed. The other parameters can be estimated through Gibbs sampling, with Metropolis-Hasting updates for the parameters λ_d and β_d. We give here the update formulas of each parameter.

For θ, one has:

$$\begin{aligned} \theta^d &\sim P(\theta | \theta^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d) \\ &= \frac{B(\alpha) B(\alpha + \lambda_d \theta^{d-1} + \Omega_d)}{B(\alpha + \Omega_d) B(\alpha + \lambda_d \theta^{d-1})} \times \\ &\quad Dir(\Omega_d + \alpha + \lambda_d \theta^{d-1}) \end{aligned} \tag{5}$$



(a) DMM (b) TTM (c) DTM (d) TM-LDA (e) ST-LDA-[D|C]

Figure 1: Graphical models for Dynamic Mixture Models (DMM, [25]), Topic Tracking Models (TTM, [11]), Dynamic Topic Models (DTM, [4]), Temporal LDA (TM-LDA, [24]) and Streaming-LDA (ST-LDA-[D|C])

where Ω_d is defined as in [23] and represents the d^{th} row of the $D \times K$ count matrix Ω, with $\Omega_{d,k}$ being the number of times that topic k is assigned to words in document d.

The update for ϕ_k^d, $1 \leq k \leq K$ is similar:

$$\begin{aligned} \phi_k^d &\sim P(\phi_k|\theta^{d-1}, \theta^d, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \mu_d) \\ &= \frac{B(\beta)B(\beta + \mu_d\phi_k^{d-1} + \Psi_k)}{B(\beta + \Psi_k)B(\beta + \mu_d\phi_k^{d-1})} \times \\ &\quad Dir(\Psi_k + \beta + \mu_d\phi_k^{d-1}) \end{aligned} \tag{6}$$

where Ψ_k is again defined as in [23] and represents the k^{th} row of a $K \times V$ count matrix, $\Psi_{k,v}$ being the number of times that topic k is assigned to word v in the documents seen so far.

The Gibbs update for z is the same as the one for the standard LDA model:

$$\forall k, 1 \leq k \leq K, P(z_v^d = k|\theta^d, \phi^d) = \frac{\theta_k^d \times \phi_{k,v}^d}{\sum_j \theta_j^d \times \phi_{j,v}^d} \tag{7}$$

Finally, for λ_d and μ_d, one can not directly compute Gibbs updates as the normalizing factor for the distribution of λ given all the other parameters can not be computed exactly. One can nevertheless rely on a Metropolis-Hasting procedure, detailed in Appendix A.

2.2 Copula-based dependencies

Model ST-LDA-D captures topic and word-topic dependencies through Dirichlet distributions, which allow one to balance the influence of the priors (α and β) and of the topic and topic-word distributions of the previous document. We introduce now another extension of LDA in which the dependencies between the topics of consecutive documents are modeled through copulas, which constitute a generic tool to model dependencies and do not rely on a specific distribution. We first provide a brief overview of copulas, prior to describe our model.

2.2.1 Basics on copulas

For every $p \geq 2$, a p–dimensional copula is a p–variate density function on $[0,1]^p$, whose univariate marginals are uniformly distributed on $[0,1]$. Copulas are particularly useful when modeling dependencies between random variables. Indeed, the joint cumulative distribution function (CDF) $F_{X_1,\cdots,X_p}$ of any random vector $\mathbf{X} = (X_1, \cdots, X_p)$ can be written as a function of its marginals, as follows:

Theorem 1 (**Sklar's theorem** Theorem 2.3.3 of [16]) *Let $F_{X_1,\cdots,X_p}$ be a p–dimensional distribution function with marginals $F_{X_1}, \cdots, F_{X_p}$. Then there exists a copula C with uniform marginals such that:*

$$F_{X_1,\cdots,X_p}(x_1, \cdots, x_p) = C(F_{X_1}(x_1), \cdots, F_{X_p}(x_p))$$

Furthermore, when the CDF $F_{X_1,\cdots,X_p}$ is continuous, the copula is unique.

Copulas represent a general way of modeling the dependencies between random variables, from complete independence to equality. If the random variables $X_1, \cdots, X_p$ are pairwise independent, their copula is the so–called *independency copula*:

$$F_{X_1,\cdots,X_p}(x_1, \cdots, x_p) = F_{X_1}(x_1) \cdots F_{X_p}(x_p)$$

whereas in the case $X_1 = \cdots = X_d$, one gets the *comonotonicity copula*:

$$F_{X_1,\cdots,X_p}(x_1, \cdots, x_p) = \min_{i \in \{1,\cdots,p\}} F_{X_i}(x_i)$$

Several copula families have been defined in the literature, among which the Archimedean copulas ([16, Ch. 4]), particularly interesting in our case. A p–dimensional Archimedean copula C with generator ψ is defined as:

$$C_p(u; \psi) := \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_p)), u \in [0,1]^p$$

where ψ is a continuous, decreasing function, from $[0, \infty]$ to $(0,1)$, strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$, and satisfying:

$$\psi(0) = 1,\ \psi(\infty) = \lim_{t \to \infty} \psi(t) = 0$$

Archimedean copulas have the following interesting properties:

- They are symmetric, that is invariant by any permutation of their coordinates, which is important when dealing with exchangeable random variables, as is the case here[2];

[2]The LDA model is based on the assumption that topics are infinitely exchangeable within a document.

- They are associative: for any $(u_1, \cdots, u_p) \in [0,1]^p$, one has:

$$C_{p-1}(C_2(u_1, u_2; \psi), u_3, \cdots, u_p; \psi)$$
$$= C_{p-1}(u, \cdots, u_{p-2}, C_2(u_{p-1}, u_p; \psi); \psi)$$

  This means that the dependency properties are the same whatever the way we group the random variables.

In this study, we further consider a particular case of the Archimedean copulas, namely the one–parameter family of Franck copula, defined, for any $\lambda \in \mathbb{R} \setminus \{0\}$, as:

$$C_\lambda(u, v) = -(1/\lambda) \ln(1 + \frac{(e^{-\lambda u} - 1)(e^{-\lambda v} - 1)}{e^{-\lambda} - 1}) \quad (8)$$

When $\lambda \to 0$, one approaches the independency copula, whereas $\lambda = \infty$ yields the comonotonicity copula. Lastly, for any $\lambda \in \mathbb{R} \setminus \{0\}$, C_λ is twice differentiable on $[0,1]^2$ so that the copula function admits a density, denoted in the sequel c_λ. By varying λ from 0 to ∞, Franck copula allows one to model all the possible dependencies between two random variables, from complete independency to equality. Dependency/independency is furthermore controlled by a single parameter, λ, which makes parameter estimation both easier and more robust.

2.2.2 Generative process

Instead of generating the topic distribution of each document θ^d independently, as is done in standard `LDA` we bind, as for our first model, `ST-LDA-D`, the topic distributions θ^{d-1} and θ^d of consecutive documents, this time by using copulas, and more precisely Franck copula.

One can not however directly use Sklar's theorem as it does not extend to joint distributions over random vectors. This means that if we are given two random vectors $\mathbf{X}_1, \mathbf{X}_2$, one can not claim that there exists a copula C such that, for any $(\mathbf{x_1}, \mathbf{x_2}) \in [0,1]^{p_1} \times [0,1]^{p_2}$:

$$F_{\mathbf{X}_1, \mathbf{X}_2}(\mathbf{x_1}, \mathbf{x_2}) = C(F_{\mathbf{X}_1}(\mathbf{x_1}), F_{\mathbf{X}_2}(\mathbf{x_2}))$$

except in very specific situation as when $\mathbf{X_1}$ and $\mathbf{X_2}$ are independent for example. One can nevertheless relate latent topics θ^{d-1} and θ_d through their components. Indeed, the topic Dirichlet distribution can be decomposed into univariate Gamma distributions with parameters $(\alpha, 1)$, denoted $Ga(\alpha)$:

Theorem 2 (from Theorem 2.1 of [17]) A random vector θ follows a Dirichlet distribution $Dir(\alpha)$ iff there exists a random vector $\mathcal{T} \sim Ga(\alpha) \otimes \cdots \otimes Ga(\alpha)$ such that:

$$\theta \overset{(\mathcal{L})}{=} \frac{\mathcal{T}}{\|\mathcal{T}\|_{\ell_1}} \quad (9)$$

where $\overset{(\mathcal{L})}{=}$ means "equality in distribution". In addition, if we are given $\theta \sim Dir(\alpha)$ and $R \sim Ga(K\alpha)$ independent, then $\mathcal{T} = R\theta \sim Ga(\alpha) \otimes \cdots \otimes Ga(\alpha)$.

To bind the topic distributions θ^{d-1} and θ^d of two consecutive documents, we thus consider the associated vectors $\mathcal{T}^{d-1}$ and $\mathcal{T}^d$, and bind them coordinate per coordinate using Franck copula. For the word-topic distributions, we use the same coupling between consecutive documents as the one used in model `ST-LDA-D`, as a tighter coupling through copulas would be too costly. We will come back to this issue in Section 3.

In the sequel for any $\gamma > 0$, f_γ (resp. F_γ) denotes the pdf (resp. cdf) of the Gamma distribution with parameters $(\gamma, 1)$. The global generative model is thus as follows:

1. Generate the first document according to the standard `LDA` model
2. For each document d, $2 \leq d \leq D$:
   (a) Generate $\lambda_d \sim U[0, \tau_\lambda]$
   (b) Generate $\mu_d \sim U[0, \tau_\mu]$
   (c) For each topic k, $1 \leq k \leq K$:
   - Generate $\mathcal{T}_k^d$ whose conditional density w.r.t. $\mathcal{T}_k^{d-1}$ is:

     $$P(\mathcal{T}_k^d | \mathcal{T}_k^{d-1}) = f_\alpha(\mathcal{T}_k^d)\, c_{\lambda_d}(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d))$$

   - Generate $\phi_k^d | \phi_k^{d-1} \sim Dir(\beta + \mu_d \phi_k^{d-1})$

   (d) Set $\theta^d = \mathcal{T}^d / \|\mathcal{T}^d\|_{\ell_1}$
   (e) For each word n, $1 \leq n \leq N$ in d:
   - Choose a topic assignment: $z_n^d \sim mult(1, \theta^d)$
   - Choose the word w_n^d from the topic z_n^d with probability $P(w_n^d | z_n^d) = \phi^d_{z_n^d, w_n^d}$

where $\mathcal{T}_k^d$ represents the k^{th} coordinate of the vector $\mathcal{T}^d$, and follows a distribution $Ga(\alpha)$ according to Theorem 2. We refer to the corresponding model as `ST-LDA-C`. Figure 1 provides a graphical representation of this model, together with the ones of previous models.

2.2.3 Inference with Gibbs sampling

The updates for z^d, ϕ^d and μ_d are identical to the ones for model `ST-LDA-D`. For λ_d, one gets:

$$P(\lambda_d | \mathcal{T}^{d-1}, \mathcal{T}^d, z^d, w^d, \alpha, \beta, \phi^{d-1}, \phi^d, \mu_d) \propto$$
$$P(\lambda_d) \prod_{k=1}^{K} f_\alpha(\mathcal{T}_k^{d-1}) f_\alpha(\mathcal{T}_k^d) c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d))$$

The same Metropolis-Hasting procedure as the one used for model `ST-LDA-D` and detailed in Appendix A can then be used.

For θ^d, one needs first to estimate the conditional probability of the random vector $\mathcal{T}^d$ with respect to the other parameters. This expression can be factored as follows:

$$P(\mathcal{T}^d | \mathcal{T}^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d) = \frac{P(\mathcal{T}^d | \mathcal{T}^{d-1}, \alpha, \lambda_d) P(z^d | \mathcal{T}^d)}{P(z^d | \alpha)}$$

As in the classical context of `LDA`, one has $P(z^d|\alpha) = B(\Omega_d + \alpha)/B(\Omega_d)$ where Ω_d is defined as before. By assumption on the distribution of the random vectors $(\mathcal{T}^{d-1}, \mathcal{T}^d)$:

$$P(\mathcal{T}^d | \mathcal{T}^{d-1}, \alpha, \lambda_d) = \prod_{k=1}^{K} f_\alpha(\mathcal{T}_k^d) c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d))$$

Developing $P(z^d|\mathcal{T}^d)$ as detailed in Appendix B, finally leads to:

$$P(\mathcal{T}^d | \mathcal{T}^{d-1}, z_d, w_d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d) \propto (\sum_{k=1}^{K} \mathcal{T}_k^d)^{-N} \times \prod_{k=1}^{K} f_{(\Omega_{d,k} + \alpha - 1)}(\mathcal{T}_k^d) \times c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d)) \quad (10)$$

Each $\mathcal{T}_k^d$ can then be estimated through the Metropolis-Hasting procedure presented in Appendix A; θ^d is finally obtained from $\mathcal{T}^d$ through Eq. 9.

3. COMPUTATIONAL CONSIDERATIONS

For model ST-LDA-C, the word-topic distributions ϕ_k^d ($1 \leq k \leq K$) could be estimated in the same way as θ^d is estimated, as mentioned in Section 2.2. However, this would entail running $K \times V$ Metropolis-Hasting procedures, which is problematic as soon as the collections considered are relatively large. We thus proposed in Section 2.2 to estimate it through Eq. 6, as done for ST-LDA-D. This time, $K \times V$ Gibbs sampling updates are required. If this estimation procedure is faster, it may still be too slow for really large collections. Theorem 2 nevertheless suggests a way to approximate ϕ_k^d ($1 \leq k \leq K$, $2 \leq d \leq D$) through Gamma updates, as follows:

1. For each word v in d, generate $t_{k,v} \sim Ga(\beta + \phi_{k,v}^{d-1})$
2. For each word v in the vocabulary $\mathcal{V}$, $\phi_{k,v}^d \leftarrow \frac{t_{k,v}}{\sum_{v \in \mathcal{V}} t_{k,v}}$

where β corresponds to the real parameter (*i.e.*, the constant value that makes up the V dimensional vector of priors). The quantities $t_{k,v}$ are first initialized through $t_{k,v} \sim Ga(\beta)$, and updated each time a new document is encountered. As one can note, this update primarily concerns the words present in the current document (step 1), the components for the other words being just renormalized (step 2). This contrasts with Eq. 6 in which the contribution of all words is resampled for each document via a multivariate Dirichlet distribution. The above procedure simplifies this by relying on the univariate equivalent of the Dirichlet distribution, namely the Gamma distribution, and by binding the variables through the renormalization step. It is faster as it involves only $K \times N$ samplings from a Gamma distribution instead of K samplings from a multivariate, $V(V >> N)$ dimensional Dirichlet distribution (the $K \times V$ renormalizations in step 2 do not really harm the procedure and are negligible compared to the Dirichlet samplings). We have observed in practice no difference, in terms of performance measures we consider (see Section 4), between this procedure and the more complex ones mentioned before, and make use of it in the remainder of the paper. In terms of speed, this procedure performed 1.5 times faster on the NIPS collection, which contains long documents and a relatively small vocabulary (*ca.* 12,000 words), and 2 times faster for the TDT4 and Tweets collections, which contain shorter documents with a larger vocabulary (up to 42,000 words).

Algorithm 1 summarizes the inference process we rely on. It makes use of the above procedure to estimate ϕ, referred to as ϕ-procedure.

Algorithm 1: Inference process for ST-LDA-[D|C]

Input: Stream of D documents of length N; number of topics K
Output: For each document d, topic distribution θ^d, word-topic distributions ϕ_k^d ($1 \leq k \leq K$); for each word v in d, topic assignment z_v^d

```
// Initialization
for k = 1 to K, v ∈ V do
    t_{k,v} ~ Ga(β)
for d = 1 to D do
    Random initialization of λ_d, μ_d and z_n^d, 1 ≤ n ≤ N
λ_1 = μ_1 = 0
// Document processing
for d = 1 to D do
    repeat
        For ST-LDA-D: update θ^d acc. to Eq. 5
        For ST-LDA-C:
            (a) update T^d (Metropolis-Hasting)
            (b) obtain θ^d from T^d through Eq. 9
        Update φ_k^d acc. φ-procedure
        Update λ_d and μ_d (Metropolis-Hasting), d > 2
        Update z_n^d acc. to Eq. 7, 1 ≤ k ≤ K, 1 ≤ n ≤ N
    until estimates are stable
```

4. EXPERIMENTAL STUDY

We conducted a number of experiments aimed at evaluating how the proposed models behave on different collections by analyzing their stability, convergence time and performance.

Datasets. We performed experiments on three datasets with different characteristics. The NIPS dataset contains 1,500 scientific papers with no time dependency between them. The size of the vocabulary is 12,375 and documents contain 500 unique words in average. The collection was collected from the NIPS proceedings and is relatively homogeneous in terms of the topics covered. It allows us to assess whether topic dependencies are still useful in a "loose" context in which there is no more temporal dependency. It is available at the UCI ML Repository [12].

The Multilingual Text and Annotations data set (TDT4)[3] proposed for topic detection and tracking, has 3,190 original documents in English and a vocabulary size of 22,965. Documents here are newswires extracted from different broadcasts and the number of unique words per document is 100 in average. Even though newswires are not extracted from the same source, they are ranked by the time.

The Tweets dataset is collected using Twitter's streaming API during 20 days from 8/10/2014 to 27/10/2014. The collection contains 72,592 tweets and a vocabulary of size 42,336. Tweets have been sequenced by time and are filtered over health issues using an SVM classifier trained over MeSH categories[4].

Each dataset was separated into training and test sets. The NIPS collection was randomly splitted into training (90% of the collection) and test (10% of the collection) sets. For TDT4, we used the first 2800 newswires released in time for training, and the last 390 ones for testing. For the Tweets dataset, we used the tweets issued in the first 17 days for training (60,000 documents) and those of the last 3 days (12,000 documents) for testing. Table 1 summarizes the characteristics of these collections.

Evaluation. Results are evaluated over the test set using the widely used perplexity measure that can be approximated by [5].

$$perplexity(C^{test}) = \exp\left(\frac{-\sum_d \sum_n \log \sum_k \theta_k^d \times \phi_{k,v_n^d}^d}{D^{test} \times N}\right) \quad (11)$$

[3]Linguistic Data Consortium, The Trustees of the University of Pennsylvania https://catalog.ldc.upenn.edu/LDC2005T16.
[4]https://www.nlm.nih.gov/mesh/

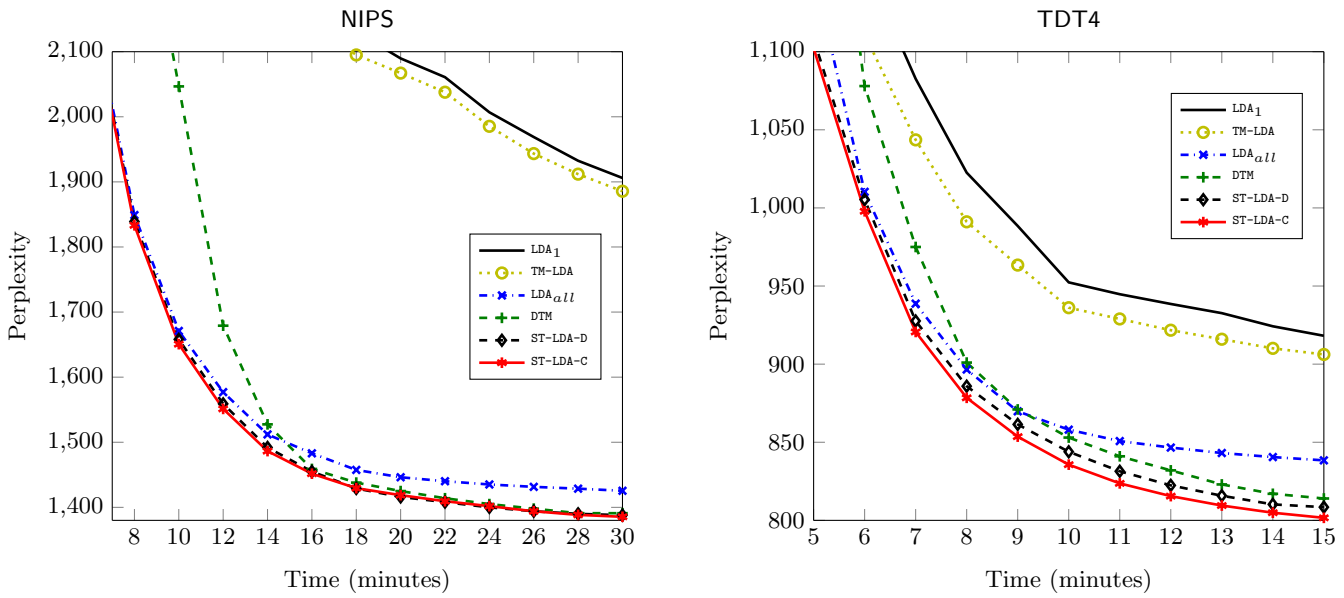


Figure 2: Perplexity curves with respect to time for all methods on NIPS and TDT4 collections (80 topics).

Table 1: Datasets used in our experiments along with their properties.

	NIPS	TDT4	Tweets
Documents in Train set	1,350	2,800	60,000
Documents in Test set	150	390	12,000
Vocabulary size	12,375	22,965	42,336
# of unique words per doc.	500	100	15
Words in total	1,900,000	779,000	904,262

where C^{test} denotes the test collection, D^{test} is its size and v_n^d represents the word at position n in document d. The parameters θ_k^d and ϕ_k^d are estimated on the training set. Furthermore, for the TDT4 collection we use the available semantic labels of newswires in the test set in order to evaluate the ability of the models to find documents of the same semantic labels using only their predicted topic distributions (Section 4.2). To this aim, we measure ROC curves and AUC of different topic models on TDT4.

Settings and comparisons. For all models, both hyperparameters α and β were fixed to 0.5. Documents of the NIPS dataset are initially stoplisted, we did not perform further preprocessing of the data nor removed stop words from the TDT4 and Tweets documents as for all methods best results are obtained when collections are not filtered.

To validate the streaming LDA models described in the previous section, we test the following six methods. The first two are LDA models [5]: (a) LDA_1, which consists in training an LDA model on the whole training data, then fixing ϕ and updating θ for each document in the test set, (b) LDA_{all}, which consists in training an LDA model on the whole on training data and updating both ϕ and θ for each document in the test set. In addition, we consider two state-of-the-art latent models that take into account dependencies between topics: Dynamic Topic Model (DTM) [4] and Temporal LDA (TM-LDA) [24]. DTM is certainly the most popular model to take into account topic dependencies. It is furthermore complete in the sense that it integrates both topic and word-topic distributions. TM-LDA is a very recent proposal with nice features. Lastly, we also consider the two streaming LDA models we have introduced (ST-LDA-D and ST-LDA-C). For these last two models, τ_λ (see Appendix A) is set to 30,000[5]. All the algorithms were implemented in Python with Numpy and Scipy[6] except DTM that is a C++ implementation tool from [3]. For both training and test, DTM is used considering that each document corresponds to a time slice.

4.1 The effect of streams of documents

We start our evaluation by analyzing the gains provided by modeling dependencies between topics by streaming (as with ST-LDA-D and ST-LDA-C) compared to other approaches on the different datasets. Figure 2 shows the evolution of perplexities of different models over the test set with respect to the training time of each model on NIPS and TDT4 datasets. The code program of DTM (in C++) generally executes faster than the other code programs (written in pyhton), nevertheless we ignore this detail and consider all the curves identically.

To measure the perplexity for each model, we estimate θ and ϕ over respectively all documents and all words of the training set. These estimates are then used to evaluate iteratively new ϕ and θ distributions for each document in the test set. This iterative update of ϕ and θ is done for all of the methods except LDA_1 which updates the distributions θ and ϕ over the whole documents in the test set with the last parameters that were obtained from the training set.

As expected, all perplexity curves decrease monotonically with respect to time. On both datasets, perplexity curves

[5]This value, upper bounding λ_d, corresponds to a regime of the Franck copula close to comonotonicity.

[6]We are working to release all the programs developed in this study publicly available for research purpose.

Table 2: Perplexity with respect to different number of topics in $\{20, 40, 60\}$.

Models	NIPS			TDT4			Tweets		
	20	40	60	20	40	60	20	40	60
LDA$_1$	2068.4	2034.5	1986.4	900.8	930.2	960.4	470.8	580.3	615.5
LDA$_{all}$	1625.4	1534.7	1458.1	723.1	768.4	792.7	431.8	508.6	577.1
TM-LDA	2038.7	2025.4	1985.3	876.7	900.3	916.3	455.1	520.1	585.2
DTM	1737.5	1551.2	1450.7	869.1	836.7	820.9	559.45	578.25	607.41
ST-LDA-D	1620.4	1520.9	1450.2	724.4	758.1	784.4	393.9	480.1	552.7
ST-LDA-C	**1612.8**	**1497.6**	**1434.5**	**720.6**	**752.5**	**780.8**	**388.2**	**474.1**	**546.8**

ST-LDA-D and ST-LDA-C lower-bound the other curves on all iterations. On the NIPS dataset, DTM becomes competitive with the two others, at the end of the iterations, while on TDT4, where test documents come in a stream, ST-LDA-C stands clearly as the best model. These results show the ability of ST-LDA-C to capture dependencies between topics in document streams. Further, we note that at the beginning of iterations where dependencies are not yet apparent, the perplexity curves of both models are very similar to the one of LDA$_{all}$. This is in line with our assertion of the previous section supporting that both models reduce to LDA in the case where topics are independent. TM-LDA is not competitive in this setting as it does really not make advantage of the fact that the words in the new, arriving documents are known. Its ability to predict future topics is not exploited in this setting.

The evolution of perplexity on Tweets from the three last consecutive days considered in our experiments is shown in Figure 3. The behavior of perplexity curves here are accentuated with the total stream characteristics of Tweets; the curve of LDA$_{all}$ gets away from those of ST-LDA-C and ST-LDA-D, while DTM comes close. In order to see if the number of topics, that we fixed for all models to 80, have an impact on these results or not, we repeated the experiments by varying the number of topics in the set $\{20, 40, 60\}$.

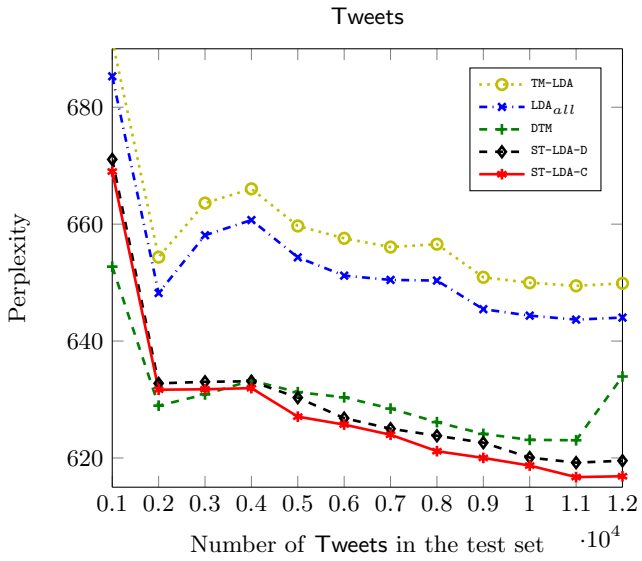


Figure 3: Perplexity of each method by number of tweets that are added to the test set (80 topics).

Table 2 depicts the perplexities of all models on the three collections at the end when the parameters ϕ and θ have been estimated over all the test documents. In all experiments, best results are obtained with ST-LDA-C and ST-LDA-D, followed by DTM on NIPS and TDT4 and by LDA$_{all}$ on Tweets. These results are consistent with those of the figures 2 and 3. Again, TM-LDA does not perform well (as explained before); LDA$_{all}$ which is a standard LDA model, performs relatively well; however, both DTM and the ST-LDA-[D|C] models outperform it by taking into account dependencies between topics. We see here that the extra flexibility of the ST-LDA-[D|C] models allow them to outperform DTM.

4.2 Ability to detect semantic correlations

We further investigate on the ability of models to find topics that can detect documents of the same semantic class. For doing so, we used the TDT4 collection for which some documents are assigned semantic classes by experts. We hence use the cosine measure or the λ_d parameter of ST-LDA-C, to detect consecutive documents in the test set of this collection that are found similar on the basis of their topic distributions; two consecutive documents are considered as similar if the cosine measure of their topic distributions (resp. estimated λ_d - line 13 Algorithm 1) is higher than a given threshold. If two consecutive and similar documents share the same semantic label, we count them as a true positive; if they do not share the same semantic label, we count them as false positive. By changing the threshold, we can plot the ROC curves for the corresponding method.

Figure 4 depicts ROC curves of DTM, TM-LDA and ST-LDA-C defined over 8 different thresholds taken in the set [0.2 0.5 0.7 0.86 0.89 0.92 0.95 0.98] for the cosine measure and [0.5 1 2 5 10 15 20 50] for λ_d when the number of topics is fixed to 20 and to 80.

In order to compare between the different ROC curves, we estimated the area under them, shown in Table 3. From these results it comes clear, that topic distributions found by ST-LDA-C are more able to detect these semantic classes than topic distributions of DTM and TM-LDA.

Table 3: Areas under the ROC curves of figure 4.

Methods	20 (Fig. 4, left)	80 (Fig. 4, right)
ST-LDA-C with λ_d	0.7982	**0.8306**
ST-LDA-C with cosine	**0.8004**	0.7755
TM-LDA with cosine	0.7652	0.7349
DTM with cosine	0.7357	0.6301

Finally, to further illustrate the role of λ_d, we pictorially illustrate the correlation between the estimated λ_d and the topic distributions of three consecutive documents (Figure 5) with identical labels in the TDT4 collection. As one can

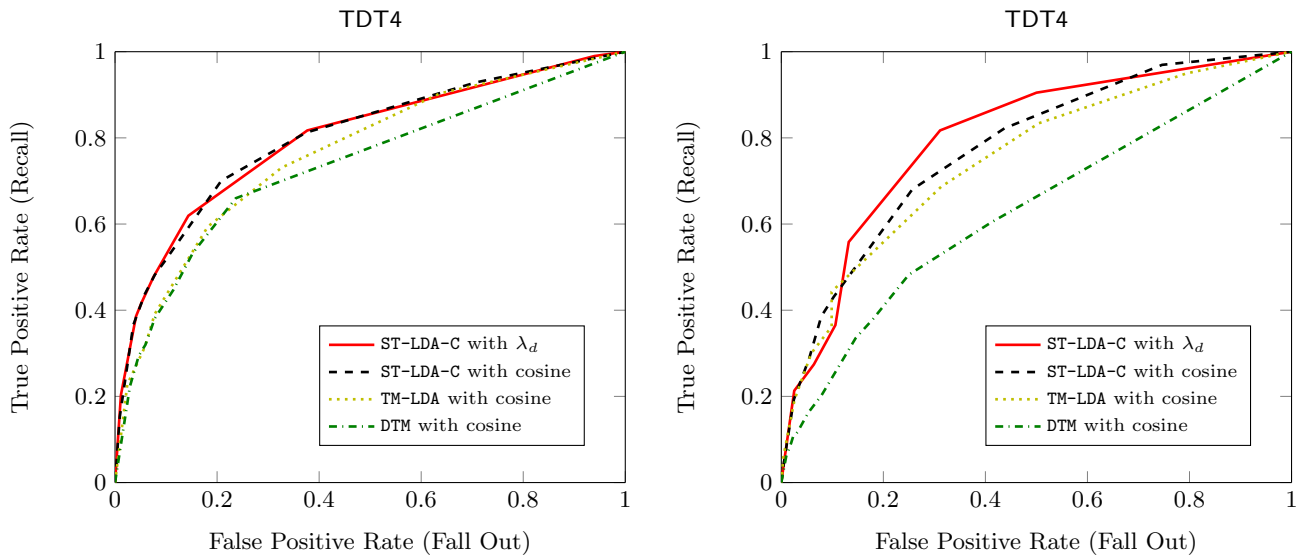


Figure 4: ROC curves of "semantic class matching" methods working over the topic distributions found by DTM, TM-LDA and ST-LDA-C, for the number of topics fixed to 20 (left) and 80 (right).

see, the distributions of topics in the three pairs of consecutive documents with high λ_d are similar. In addition, the two most probable topics of the document pairs retained in Figure 6, also taken from TDT4, do not share any word when λ_d is small and are almost identical when λ_d is high. These examples illustrate the fact that λ_d is a good indicator of the topic dependencies between documents.

5. RELATED WORK

Some studies have considered the possibility to model different streams of documents, as in [10], trying to leverage standard models (as LDA) by considering topics common to the different streams. In such studies the evolution of topics over time is not considered. The study presented in [22] aims at modeling, through an extension of LDA, the timestamp associated with each token in a document. If dependencies between topics are not explicitly modeled, topics tend to specialize over different time periods through the joint dependence of each word and timestamp on the topic variable (z in LDA). Other studies have addressed the problem of topic evolution and dependencies within a single document, as the recent *sequential* LDA model described in [7]. We rather focus in this study on explicitly modeling topic dependencies across documents, for both topic and word-topic distributions. Several studies have addressed a similar problem. One of the first proposals corresponds to the Dynamic Topic Model (DTM), introduced in [4] and illustrated in Figure 1. An interesting feature of DTM is its use of time slices; we have not considered time slices in this study, but our models (as most dynamic models) can be extended to deal with them. DTM captures dependencies for both topic and word-topic distributions. These dependencies are however captured through Gaussian distributions, the expectation of which corresponds to the previous parameters. This entails that new parameter values are constrained to be distributed around the values observed previously. In contrast, even in model ST-LDA-D, the expectations of the new topic and word-topic distributions (Eqs. 2 and 4) can be uncorrelated to the previous distributions in the absence of dependencies. Our models thus offer additional flexibility over the presence or absence of dependencies between consecutive documents in a stream. The Dynamic Mixture Model (DMM, see Fig.1) introduced in [25] is similar to DTM except that topic dependencies are directly considered at the topic level (as is the case for ST-LDA-D and ST-LDA-C but not for DTM which operates at the prior level) and that word-topic dependencies are dropped. As for DTM, the expectation of a new topic distribution is given by the values obtained in the previous document. This again contrasts with our proposal that introduces additional flexibility, as mentioned before. The Topic Tracking Model (TTM, see Fig.1) introduced in [11] is similar to our models in the sense that both topic and word-topic (more precisely interest-topic) dependencies are considered. However, as for DTM and DMM, the mean of the current topics and interests are the same as the ones of the previous topics and interests. The model is thus again limited in its ability to model the presence or absence of dependencies between consecutive documents. A more recent proposal, called Temporal LDA (TM-LDA, see Fig.1), was introduced in [24]. TM-LDA differs from the previous models as it also aims at predicting future topics even in the situation where future documents are not seen. It thus assumes a strong dependency between consecutive documents, which is not always realistic, even on such collections as Tweets. Furthermore, TM-LDA does not consider dependencies for the word-topic distributions.

6. CONCLUSION

We have proposed in this paper two new models for modeling topic and word-topic dependencies between consecutive documents in document streams. The first model is a direct extension of Latent Dirichlet Allocation model (LDA) and makes use of a Dirichlet distribution to balance the influence of the LDA prior parameters wrt to topic and word-topic distribution of the previous document. The second extension makes use of copulas, which constitute a generic

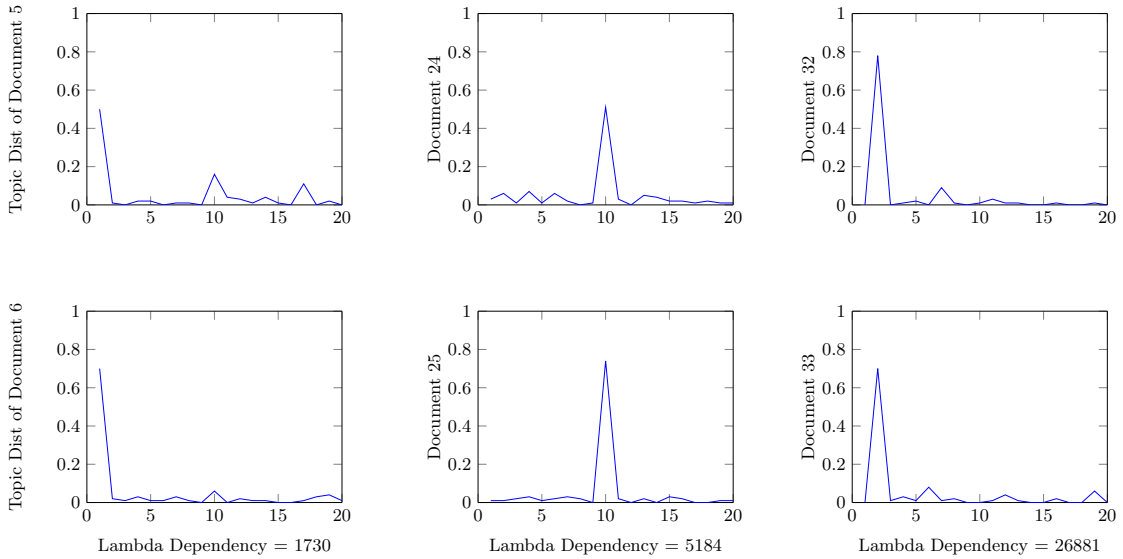


Figure 5: Topic distribution of three pairs consecutive documents that have the same topic (*Olympic* - left, *Election* - middle, *Sport* - right) and subject labels in TDT4 dataset (20 topics).

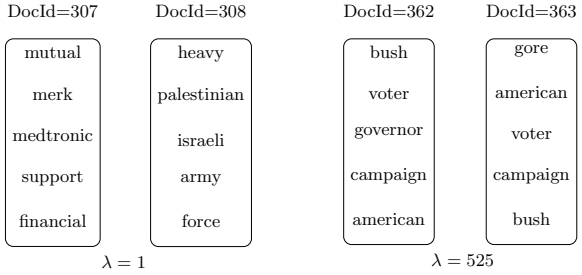


Figure 6: 5 most frequent words of the most probable topic (20 topics).

tool to model dependencies between random variables. Our experiments, conducted on three standard collections that have been used in several studies on topic modeling, show that our proposals outperform previous ones (as dynamic topic models and temporal LDA), both in terms of perplexity and for tracking similar topics in a document streams. Compared to previous proposals, our models have extra flexibility and can adapt to situations where there is in fact no dependencies between the documents.

In the future, we plan to develop non-parametric extensions as well as versions of these models that scale well, following the improvements on the inference methods for LDA, proposed in streams [26] or in online settings [9, 2].

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. This work was partly supported by the LabEx PERSYVAL-Lab ANR-11-LABX-0025.

8. REFERENCES

[1] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI*, 2009.

[2] A. Banerjee and S. Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *Proceedings of the 7th SIAM conference on Data Mining, SDM*, 2007.

[3] D. M. Blei. Free C++ implementation for dtm. https://www.cs.princeton.edu/~blei/topicmodeling.html.

[4] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ACM International Conference Proceeding Series, ICML*, 2006.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[6] S. Derrode and W. Pieczynski. Unsupervised data classification using pairwise markov chains with automatic copulas selection. *Computational Statistics & Data Analysis*, 2013.

[7] L. Du, W. L. Buntine, and H. Jin. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. In *IEEE Computer Society, ICDM*, 2010.

[8] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: A review. *ACM SIGMOD Record*, 2005.

[9] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *NIPS*, 2010.

[10] L. Hong, B. Dom, S. Gurumurthy, and K. Tsioutsiouliklis. A time-dependent topic model for multiple text streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2011.

[11] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda. Topic tracking model for analyzing consumer purchase behavior. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence, IJCAI*, 2009.

[12] M. Lichman. UCI machine learning repository, 2013.

[13] A. J. McNeil. Sampling nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, 2008.

[14] A. J. McNeil and J. Nešlehovà. Multivariate Archimedean copulas, D-monotone functions and ℓ_1-norm symmetric distributions. *Annals of Statistics*, 2009.

[15] R. Neal. Slice sampling. *Annals of Statistics*, 2000.

[16] R. B. Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

[17] K. W. Ng, G.-L. Tian, and M.-L. Tang. *Dirichlet and related distributions: Theory, methods and applications*. John Wiley & Sons, 2011.

[18] O. Ostap, O. Yarema, and S. Wolfgang. Properties of hierarchical Archimedean copulas. *Statistics & Risk Modeling*, 2013.

[19] M. J. Paul and M. Dredze. You are what you tweet: Analyzing twitter for public health. In *International Conference on Weblogs and Social Media*, 2011.

[20] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *International Conference on World Wide Web*, 2010.

[21] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking LDA: why priors matter. In *Advances in Neural Information Processing Systems Conference*, 2009.

[22] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2006.

[23] Y. Wang. Distributed gibbs sampling of latent topic models: The gritty details. Technical report, 2008.

[24] Y. Wang, E. Agichtein, and M. Benzi. TM-LDA: Efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2012.

[25] X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time series. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI*, 2007.

[26] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2009.

APPENDIX

A. METROPOLIS-HASTING PROCEDURE

The Metropolis-Hasting procedure is based on the following steps:

1. Generate an initial value of x: draw $x^1 \sim P_{\text{prior}}(x)$
2. Initialize $j = 1$
3. Repeat till sequence is stable
   (a) Draw $x \sim q$, where q represents the "jump" function
   (b) Draw $u \sim U[0,1]$
   (c)

$$\alpha = \begin{cases} \frac{\Pi(x^j)q(x)}{\Pi(x)q(x^j)} & \text{if } \Pi(x^j)q(x) < \Pi(x)q(x^j) \\ \frac{\Pi(x)q(x^j)}{\Pi(x^j)q(x)} & \text{otherwise} \end{cases}$$

   (d) If $u \leq \alpha$, then $x^{j+1} = x$; $x^{j+1} = x^j$ otherwise

For $x = \lambda_d$, one has:

$$\begin{aligned} & P(\lambda_d|\theta^{d-1}, \theta^d, z^d, w^d, \alpha, \beta, \phi^{d-1}, \phi^d, \mu_d) \\ \propto \quad & P_{\text{prior}}(\lambda_d) P(\theta^d|\theta^{d-1}, \alpha, \lambda_d) \ := \Pi(\lambda_d) \end{aligned}$$

where $P_{\text{prior}}(\lambda_d) \sim U[0, \tau_\lambda]$. As λ_d should be higher when θ^{d-1} and θ^d are more similar (as in such a case the influence of θ^{d-1} on θ^d is more important), we make use of the following jump function, based on the exponential distribution:

$$q(\lambda_d) = (1 - cos(\theta^{d-1}, \theta^d)) \times e^{-(1-cos(\theta^{d-1}, \theta^d)) \times \lambda_d}$$

For $x = \mu_d$, the same distribution is used for the jump function, the cosine being taken between the vectors that correspond to the column-wise concatenation of the columns of each matrix ϕ^{d-1} and ϕ^d. The prior this time is $P(\mu_d) \sim U[0, \tau_\mu]$. Lastly, for $x = \mathcal{T}_k^d$, $P_{\text{prior}}(\mathcal{T}_k^d) \sim Ga(\alpha)$, the jump function corresponds to Franck copula, and $\Pi(\mathcal{T}_k^d)$ corresponds to the k^{th} contribution in Eq. 10.

B. GIBBS SAMPLING UPDATES (*ST-LDA-C*)

We provide here the complete derivation of Eq. 10. For any $d \geq 2$, one has:

$$\begin{aligned} \mathcal{T}^d \quad \sim \quad & P(\mathcal{T}^d|\mathcal{T}^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d) \\ = \quad & \frac{P(\mathcal{T}^{d-1}|\alpha) P(\mathcal{T}^d|\mathcal{T}^{d-1}, \alpha, \lambda_d) P(z^d|\mathcal{T}^d) P(w^d|z^d)}{P(\mathcal{T}^{d-1}|\alpha) p(z^d|\alpha) P(w^d|z^d)} \\ = \quad & \frac{P(\mathcal{T}^d|\mathcal{T}^{d-1}, \alpha, \lambda_d) P(z^d|\mathcal{T}^d)}{P(z^d|\alpha)} \end{aligned}$$

Let F_α (resp f_α) denote the cdf (resp pdf) of the Gamma distribution with parameters $(\alpha, 1)$. By assumption:

$$P(\mathcal{T}^d|\mathcal{T}^{d-1}, \alpha, \lambda_d) = \prod_{k=1}^{K} f_\alpha(\mathcal{T}_k^d) c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d))$$

and, since $\theta^d = \mathcal{T}^d / \left(\sum_{k=1}^K \mathcal{T}_k^d\right)$,:

$$P(z^d|\mathcal{T}^d) = \prod_{n=1}^{N} \theta_{z_n^d}^d = \left(\sum_{k=1}^{K} \mathcal{T}_k^d\right)^{-N} \prod_{n=1}^{N} \mathcal{T}_{z_n^d}^d$$

Further, as usual [23]:

$$P(z^d|\alpha) = \int P(z^d|\theta^d) P(\theta^d|\alpha) d\theta^d = \frac{B(\Omega_d + \alpha)}{B(\Omega_d)}$$

Hence:

$$\begin{aligned} p(\mathcal{T}^d|\mathcal{T}^{d-1}, z^d, \cdots) = & \frac{\left(\sum_{k=1}^K \mathcal{T}_k^d\right)^{-N} \prod_{n=1}^N \mathcal{T}_{z_n^d}^d}{\left[\prod_{k=1}^K \Gamma(\alpha)\right] B(\Omega_d + \alpha)/B(\Omega_d)} \times \\ & \left[\prod_{k=1}^{K} \mathcal{T}_k^{d\alpha-1} \exp^{-\mathcal{T}_k^d} c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d))\right] \\ = & \frac{\left(\sum_{k=1}^K \mathcal{T}_k^d\right)^{-N} \prod_{k=1}^K \mathcal{T}_k^{d\Omega_{d,k}+\alpha-1}}{\left[\prod_{k=1}^K \Gamma(\alpha)\right] B(\Omega_d + \alpha)/B(\Omega_d)} \times \\ & \exp^{-\sum_{k=1}^K \mathcal{T}_k^d} \prod_{k=1}^{K} c_\lambda(F_\alpha(\mathcal{T}_k^{d-1}), F_\alpha(\mathcal{T}_k^d)) \end{aligned}$$

leading to the desired result.