
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 especially 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
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 $\{1, \ldots, K\}$, are used to represent the hidden (in the sense non-observed) topics underlying each document. LDA is associated to the following generative model:\footnote{For simplification and following standard practice, we do not model here the length of each document, assumed to be fixed and equal to $N$.}:

- Generate, for each topic $k, 1 \leq k \leq K$, a distribution over the words: $\phi_k \sim \text{Dir}(\beta)$, where $\phi_k$ and $\beta$ are $V$ dimensional vectors;
- For each document $d$:
  - Choose a distribution over the topics: $\theta^d \sim \text{Dir}(\alpha)$, where $\theta^d$ and $\alpha$ are $K$ dimensional vectors;
  - For each position (indexed by $n, 1 \leq n \leq N$) in $d$:
    (a) Choose a topic assignment: $z^n_d \sim \text{mult}(1, \theta^d)$;
    (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 $\phi_k$. $\alpha$ and $\beta$ 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: $\alpha_1 = \ldots = \alpha_K$. If one assumes a broad Gamma prior for both $\alpha$ and $\beta$, 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 $\alpha$ and $\beta$ 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.

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 $\alpha$ and the topic distribution estimated in the previous document:

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

where $\lambda_d$ 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 $\theta^d$ is given by:

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

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

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:

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

Here $\mu_d$ is again a uniformly distributed parameter that controls the tradeoff between the prior $\beta$ and the learned topic-word distributions $\phi^{d-1}$. As usual $\phi_k^{d-1}$ is the word distribution of topic $k$. The conditional mean of each component of $\phi_k^d$ is given by:

$$E[\phi_k^d | \phi^{d-1}] = \frac{\beta + \mu_d \phi_k^{d-1}}{\beta + \mu_d}$$ (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 $\alpha$ and $\beta$ are considered fixed. The other parameters can be estimated through Gibbs sampling, with Metropolis-Hasting updates for the parameters $\lambda_d$ and $\beta_d$. We give here the update formulas of each parameter.

For $\theta$, one has:

$$\theta^d \sim P(\theta | \theta^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d)$$

$$\propto B(\alpha)B(\alpha + \lambda_d \theta^{d-1}) \times B(\alpha + \Omega_d)B(\alpha + \lambda_d \theta^{d-1}) \times \text{Dir}(\Omega_d + \alpha + \lambda_d \theta^{d-1})$$ (5)
Indeed, the joint cumulative distribution function (CDF), density function on \([0, 1]^m\), is defined as in \([23]\) and represents the \(d\)th row of the \(D \times K\) count matrix \(\Omega\), with \(\Omega_{d,k}\) being the number of times that topic \(k\) is assigned to word \(d\) in the documents seen so far.

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

\[
\forall k, 1 \leq k \leq K, P(z^d_k = k | \theta^d, \phi^d) = \frac{\theta^d_k \times \phi^d_{k,v}}{\sum_j \theta^d_j \times \phi^d_{j,v}}
\]

Finally, for \(\lambda_d\) and \(\mu_d\), one can not directly compute Gibbs updates as the normalizing factor for the distribution of \(\lambda\) given all the other parameters can not be computed exactly. One can nevertheless rely on a Metropolis-Hastings 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 (\(\alpha\) and \(\beta\)) 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 distribution 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) \(F_{X_1,\ldots,X_p}\) of any random vector \(\mathbf{X} = (X_1, \ldots, X_p)\) can be written as a function of its marginals, as follows:

**Theorem 1 (Sklar’s theorem)** Let \(F_{X_1,\ldots,X_p}\) be a \(p\)-dimensional distribution function with marginals \(F_{X_1,\ldots,X_p}\). Then there exists a copula \(C\) with uniform marginals such that:

\[
F_{X_1,\ldots,X_p}(x_1, \ldots, x_p) = C(F_{X_1}(x_1), \ldots, F_{X_p}(x_p))
\]

Furthermore, when the CDF \(F_{X_1,\ldots,X_p}\) 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 \(X_1, \ldots, X_p\) are pairwise independent, their copula is the so-called independency copula:

\[
F_{X_1,\ldots,X_p}(x_1, \ldots, x_p) = F_{X_1}(x_1) \cdots F_{X_p}(x_p)
\]

whereas in the case \(X_1 = \cdots = X_d\), one gets the comonotonicity copula:

\[
F_{X_1,\ldots,X_p}(x_1, \ldots, x_p) = \min_{i \in \{1, \ldots, p\}} F_{X_i}(x_i)
\]

Several copula families have been defined in the literature, among which the Archimedean copulas \((16, \text{ Ch. 4})\), particularly interesting in our case. A \(p\)-dimensional Archimedean copula \(C\) with generator \(\psi\) is defined as:

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

where \(\psi\) 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:
\[
C_{p-1}(C_2(u_1, u_2); \psi), u_3, \cdots, u_p; \psi)
= C_{p-1}(u_1, \cdots, u_{p-2}, C_2(u_{p-1}, u_p; \psi); \psi)
\]
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 Frank copula, defined, for any \(\lambda \in \mathbb{R} \setminus \{0\}\), as:
\[
C_\lambda(u,v) = -(1/\lambda) \ln(1 + (e^{-\lambda u} - 1)(e^{-\lambda v} - 1)/e^{-\lambda} - 1)
\]
When \(\lambda \to 0\), one approaches the independency copula, whereas \(\lambda = \infty\) yields the comonotonicity copula. Lastly, for any \(\lambda \in \mathbb{R} \setminus \{0\}\), \(C_\lambda\) is twice differentiable on \([0,1]^2\) so that the copula function admits a density, denoted in the sequel \(c_\lambda\). By varying \(\lambda\) from 0 to \(\infty\), Frank copula allows one to model all the possible dependencies between two random variables, from complete independency to equality. Dependancy/independency is furthermore controlled by a single parameter, \(\lambda\), which makes parameter estimation both easier and more robust.

2.2.2 Generative process

Instead of generating the topic distribution of each document \(\theta^d\) independently, as is done in standard LDA we bind, as for our first model, ST-LDA-D, the topic distributions \(\theta^d-1\) and \(\theta^d\) of consecutive documents, this time by using copulas, and more precisely Frank 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 \(X_1, X_2\), one can not claim that there exists a copula \(C\) such that, for any \((x_1, x_2) \in [0,1]^p \times [0,1]^p\):
\[
F_{X_1,X_2}(x_1, x_2) = C(F_{X_1}(x_1), F_{X_2}(x_2))
\]
except in very specific situation as when \(X_1\) and \(X_2\) are independent for example. One can nevertheless relate latent topics \(\theta^d-1\) and \(\theta^d\) through their components. Instead, 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 \(\theta\) follows a Dirichlet distribution \(Dir(\alpha)\) iff there exists a random vector \(T \sim Ga(\alpha) \otimes \cdots \otimes Ga(\alpha)\) such that:
\[
\theta \overset{\text{d}}{=} \frac{T}{\|T\|_{\ell_1}}
\]
where \(\overset{\text{d}}{=}\) means “equality in distribution”. In addition, if we are given \(\theta \sim Dir(\alpha)\) and \(R \sim Ga(K\alpha)\) independent, then \(T = R\theta \sim Ga(\alpha) \otimes \cdots \otimes Ga(\alpha)\).

To bind the topic distributions \(\theta^d-1\) and \(\theta^d\) of two consecutive documents, we thus consider the associated vectors \(T^d-1\) and \(T^d\), and bind them coordinate per coordinate using Frank 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 \(\gamma > 0\), \(f_\gamma\) (resp. \(F_\gamma\)) 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 \(\lambda_d \sim U[0, \tau]\)
   (b) Generate \(\mu_d \sim U[0, \tau]\)
   (c) For each topic \(k\), \(1 \leq k \leq K\):
      • Generate \(T_k^d\) whose conditional density w.r.t. \(T_k^{d-1}\) is:
        \[
P(T_k^d | T_k^{d-1}) = f_\alpha(T_k^d) c_\lambda(F_\alpha(T_k^{d-1}), F_\alpha(T_k^d))
        \]
      • Generate \(\phi_k^d|\theta^d-1 \sim Dir(\beta + \mu_d \theta_k^d-1)\)
   (d) Set \(\theta^d = \frac{T^d}{||T^d||_{\ell_1}}\)
   (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 \(P(w_n^d | z_n^d) = \phi_{z_n^d}^d w_n^d\)

where \(T_k^d\) represents the \(k\)th coordinate of the vector \(T^d\), and follows a distribution \(Ga(\alpha)\) according to Theorem 9.

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 \(z_n^d\), \(\phi^d\) and \(\mu_d\) are identical to the ones for model ST-LDA-D. For \(\lambda_d\), one gets:
\[
P(\lambda_d | T^d, z_n^d, w_n^d, \alpha, \beta, \theta^d-1, \phi^d, \mu_d) \propto\]
\[
P(\lambda_d) \prod_{k=1}^K f_\alpha(T_k^d-1) f_\alpha(T_k^d) c_\lambda(F_\alpha(T_k^{d-1}), F_\alpha(T_k^d))
\]
The same Metropolis-Hasting procedure as the one used for model ST-LDA-D and detailed in Appendix A can then be used.

For \(\theta^d\), one needs first to estimate the conditional probability of the random vector \(T^d\) with respect to the other parameters. This expression can be factored as follows:
\[
P(T^d | T^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^d-1, \phi^d, \mu_d) =\]
\[
P(T^d | T^{d-1}, \alpha, \lambda_d) P(z^d | T^d)\]
As in the classical context of LDA, one has \(P(z^d | \alpha) = B(\Omega_d + \alpha)/B(\Omega_d)\) where \(\Omega_d\) is defined as before. By assumption on the distribution of the random vectors \((T^{d-1}, T^d)\):
\[
P(T^d | T^{d-1}, \alpha, \lambda_d) \equiv\]
\[
\prod_{k=1}^K f_\alpha(T_k^d-1) c_\lambda(F_\alpha(T_k^{d-1}), F_\alpha(T_k^d))
\]

Developing \(P(z^d | T^d)\) as detailed in Appendix B, finally leads to:
\[
P(T^d | T^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^d-1, \phi^d, \mu_d) \propto\]
\[
\prod_{k=1}^K f_{(\Omega_d + \alpha - 1)}(T_k^d) \times c_\lambda(F_\alpha(T_k^{d-1}), F_\alpha(T_k^d)) (10)
\]
Each \(T_k^d\) can then be estimated through the Metropolis-Hasting procedure presented in Appendix A; \(\theta^d\) is finally obtained from \(T^d\) through Eq. 9.
3. COMPUTATIONAL CONSIDERATIONS

For model ST-LDA-C, the word-topic distributions \( \phi_{k,v}^d \) (\( 1 \leq k \leq K \)) could be estimated in the same way as \( \theta^d \) 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 \( \phi_{k,v}^d \) (\( 1 \leq k \leq K, 2 \leq d \leq D \)) 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 \( V \), \( \phi_{k,v}^d \leftarrow \frac{t_{k,v}}{\sum_{v \in V} t_{k,v}} \)

where \( \beta \) 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.

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

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 where 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 2,800 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].

\[
\text{perplexity}(C^{\text{test}}) = \exp \left( -\frac{\sum_d \sum_n \log \sum_k \phi_{k,v}^d \times \phi_{k,v}^d}{D^{\text{test}} \times N} \right) \tag{11}
\]


\(^4\)https://www.nlm.nih.gov/mesh/
where $C^{text}$ denotes the test collection, $D^{text}$ is its size and $v^d_n$ represents the word at position $n$ in document $d$. The parameters $\theta_k^d$ and $\phi_k^d$ 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 of newswires in the test set in order to evaluate thermore, for the TDT4 hyperparameters $\alpha$ where $C$ are fixed to 0
5. Documents of the test collection, which consists in training an LDA model on the whole training data, then fixing $\phi$ and updating $\theta$ 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 $\phi$ and $\theta$ 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, $\tau_\lambda$ (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 python), nevertheless we ignore this detail and consider all the curves identically.

To measure the perplexity for each model, we estimate $\theta$ and $\phi$ over respectively all documents and all words of the training set. These estimates are then used to evaluate iteratively new $\phi$ and $\theta$ distributions for each document in the test set. This iterative update of $\phi$ and $\theta$ is done for all of the methods except LDA$_1$ which updates the distributions $\theta$ and $\phi$ 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

<table>
<thead>
<tr>
<th>NIPS</th>
<th>TDT4</th>
<th>Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents in Train set</td>
<td>1,350</td>
<td>2,800</td>
</tr>
<tr>
<td>Documents in Test set</td>
<td>150</td>
<td>390</td>
</tr>
<tr>
<td>Vocabulary size</td>
<td>12,375</td>
<td>22,965</td>
</tr>
<tr>
<td># of unique words per doc.</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Words in total</td>
<td>1,900,000</td>
<td>779,000</td>
</tr>
</tbody>
</table>

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

collection at the end when the parameters \( \phi \) and \( \theta \) 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 \( \lambda_d \) 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 \( \lambda_d \) - 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 \( \lambda_d \) 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>20 (Fig. 4, left)</th>
<th>80 (Fig. 4, right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-LDA-C with ( \lambda_d )</td>
<td>0.7982</td>
<td>0.8306</td>
</tr>
<tr>
<td>ST-LDA-C with cosine</td>
<td>0.8004</td>
<td>0.7755</td>
</tr>
<tr>
<td>TM-LDA with cosine</td>
<td>0.7652</td>
<td>0.7349</td>
</tr>
<tr>
<td>DTM with cosine</td>
<td>0.7357</td>
<td>0.6301</td>
</tr>
</tbody>
</table>

Finally, to further illustrate the role of \( \lambda_d \), we pictorially illustrate the correlation between the estimated \( \lambda_d \) 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 $\lambda_d$ 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 $\lambda_d$ is small and are almost identical when $\lambda_d$ is high. These examples illustrate the fact that $\lambda_d$ 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 uncor-related 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 (TM, see Fig.1), was introduced in [21], and that 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
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


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) \( \alpha = \begin{cases} \frac{P(x^0)q(x)}{P(x)q(x^0)} & \text{if } P(x^0)q(x) < P(x)q(x^0) \\ 1 & \text{otherwise} \end{cases} \)
   (d) If \( u \leq \alpha \), then \( x^{j+1} = x; x^{j+1} = x^b \) otherwise

For \( x = \lambda_d \), one has:
\[
P(\lambda_d | \theta^{d-1}, \theta^d, z^d, w^d, \alpha, \beta, \phi^{d-1}, \phi^d, \mu_d) \propto P_{\text{prior}}(\lambda_d) P(\theta^d | \theta^{d-1}, \alpha, \lambda_d) \equiv \Pi(\lambda_d)
\]
where \( P_{\text{prior}}(\lambda_d) \sim U[0, \tau_d] \). As \( \lambda_d \) should be higher when \( \theta^{d-1} \) and \( \theta^d \) are more similar (as in such a case the influence of \( \theta^{d-1} \) on \( \theta^d \) 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 \( \phi^{d-1} \) and \( \phi^d \). The prior this time is \( P(\mu_d) \sim U[0, \tau_d] \). Lastly, for \( x = T_k^d \), \( P_{\text{prior}}(T_k^d) \sim G(\alpha) \), the jump function corresponds to Franck copula, and \( \Pi(T_k^d) \) 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:
\[
\mathcal{T}^d \sim P(T^d | T^{d-1}, z^d, w^d, \alpha, \beta, \lambda_d, \phi^{d-1}, \phi^d, \mu_d) = \frac{P(T^d | T^{d-1}, \alpha, \lambda_d) P(z^d | T^d) P(w^d | z^d)}{P(T^{d-1} | \alpha) P(z^d | T^d) P(w^d | z^d)} = \frac{P(T^d | T^{d-1}, \alpha, \lambda_d) P(z^d | T^d)}{P(z^d | \alpha)}
\]

Let \( F_\alpha \) (resp. \( f_\alpha \)) denote the cdf (resp pdf) of the Gamma distribution with parameters \((\alpha, 1)\). By assumption:
\[
P(T^d | T^{d-1}, \alpha, \lambda_d) = \prod_{k=1}^{K} f_\alpha(T_k^d) c_\lambda(F_\alpha(T_k^d), F_\alpha(T_k^d))
\]
and, since \( \theta^d = T^d / \left( \sum_{k=1}^{K} T_k^d \right) \):
\[
P(z^d | T^d) = \prod_{n=1}^{N} \theta_n^{d_n} = \left( \prod_{k=1}^{K} T_k^d \right)^{-N} \prod_{n=1}^{N} T_n^d
\]

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:
\[
p(T^d | T^{d-1}, z^d, \ldots) = \prod_{k=1}^{K} T_k^d \prod_{n=1}^{N} T_n^d \prod_{k=1}^{K} \Gamma(\alpha) B(\Omega_d + \alpha) / B(\Omega_d)
\]
leading to the desired result.