Ringraziamenti

Grazie ai miei genitori, Chiara e Luca, per essersi sempre spesi per me, in tutti i sensi, affinché io fossi libero di fare ciò che mi rende felice: questa laurea è per voi.

A Tommaso, mio fratello, per condividere con me la vita in famiglia, e ultimamente un po' anche al di fuori di essa.

Ai miei nonni, nonne, e zia, che da sempre credono in me più di me stesso.

Grazie ai miei amici Edoardo, Andrea, e Leonardo, con cui ho condiviso i miei più bei momenti, e con cui ne vorrei vivere molti altri.

Ai miei compagni animatori: Anna, Lorena, Tommaso, e Clara, compagni immancabili dai tempi dell'oratorio, e spero anche di quelli che verranno.

Ai miei compagni di viaggio: Tommaso, Tomas, Vittorio, e Riccardo. Senza di loro questi tre ultimi anni sarebbero passati più lentamente, e più difficilmente.

Grazie infine al mio prof. Antonio Lijoi, per essere stato mio prezioso supervisore, non solo durante la stesura della tesi.

Contents

1 THE IMPORTANCE OF TOPIC MODELING

In today's age, the volume of data generated daily is overwhelming, thus making it increasingly challenging to extract meaningful insights efficiently. Inundated by this deluge of information, the importance of topic modeling emerges as a crucial tool for structuring and understanding vast corpora of text, as it allows us to unveil hidden themes and patterns.

Indeed, statistical learning techniques facilitate the arduous action of categorizing and distilling textual content into coherent and interpretable clusters: models identify structures, measure the strength of the relationships and estimate inference on the data, finding application in different fields.

In the realm of natural language processing, topic models have become indispensable tools for tasks such as document clustering and, more generally, information retrieval. For instance, they help organizing vast collections of research papers, facilitating the access to relevant literature based on thematic content rather than keyword matching alone.

Their focus on the connections and relationships between entities rather than on the entities themselves provides a unique approach to analyse complex systems of any kind, though they were first conceived as text-mining tools. In genetics, for example, they have been employed to identify patterns in gene expression data, leading to breakthroughs in understanding genetic functions and disease mechanisms. Similarly, they are used to detect recurring visual patterns and themes in images, enabling automated annotation and organization.

With data growing in both scale and complexity, the role of topic modeling becomes crucial. Statistical models in the field provide a systematic and rigorous framework that is essential not only in distilling large volumes of unstructured data into meaningful, interpretable themes, but also opens up new avenues for discovery and innovation across a wide range of disciplines by testing hypothesis.

Despite the vast potential, topic methods are complex and pose significant challenges, and, at times, the distributions used to model the texts examined are intractable, especially during the inference phase. For this reason, we need to use approximations with the scope of simplifying the computation, without sacrificing too much efficiency.

This thesis will focus precisely on these methods. After refreshing some necessary statistical fundamentals and defining the state-of-the-art models, the main inference algorithms proposed in the literature will be explored; additionally, a practical application on real-world data will be used to show the effectiveness of one them on the two models presented.

The ultimate scope of this work is to analyze and compare different inference algorithms in the context of topic modeling, providing a robust methodological framework. By explaining in a clear way the statistical fundamentals, evaluating well-known methods and their variation, and testing their application to tangible data, my goal is that of producing a comprehensive review to not only approach, but also understand exhaustively the difficulties related to topic models and how to overcome them.

2 STATISTICAL FOUNDATIONS

2.1 Dirichlet Distribution

The Dirichlet is a family of continuous distributions parameterized by a vector of positive real numbers and can be thought as a multidimensional Beta over the unit simplex, that is, a distribution over distributions.

Given $\theta = (\theta_1, \dots, \theta_n)$ and $\alpha = (\alpha_1, \dots, \alpha_n)$, we say that $\theta \sim \mathrm{Dir}(\alpha)$ iff:

$$
\begin{aligned}
p(\theta) &= \frac{1}{\mathcal{B}(\alpha)} \prod_{k=1}^{n} \theta_k^{\alpha_k - 1} \mathbb{1}\{\theta_i \in S\} \qquad\qquad S = \{\theta \in \mathbb{R}^n : \theta_i \geq 0, \sum_{i=1}^{n} \theta_i = 1\} \\
&= \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{k=1}^{n} \theta_k^{\alpha_k - 1} \mathbb{1}\{\theta_i \in S\}
\end{aligned}
$$

The support of this distribution is the set of n-dimensional probability vectors, i.e. those vectors with entries in the interval $[0, 1]$ whose sum is 1. Technically, the density of the Dirichlet is defined over a simplex, which is, by definition, $(n-1)$ dimensional, and it usually serves as a prior in Bayesian statistics, due to its conjugacy with the multinomial distribution, which will turn extremely useful during inference.

From $\theta \sim \mathrm{Dir}(\alpha)$, it follows that $\mathbf{x}|\theta \sim \mathrm{Multinomial}(\theta, \alpha)$ and we get:

$$
\begin{aligned}
p(\theta) &= \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{k=1}^{n} \theta_k^{\alpha_k - 1} \\
p(x_1, \dots, x_n|\theta) &= \frac{(\sum_{i=1}^{n} \alpha_i)!}{\prod_{i=1}^{n} x_i!} \prod_{k=1}^{n} \theta_k^{x_k} \\
\Rightarrow p(\theta|x_1, \dots, x_n) &\propto p(x_1, \dots, x_n|\theta) p(\theta) \\
&\propto \prod_{k=1}^{n} \theta_k^{x_k} \prod_{k=1}^{n} \theta_k^{\alpha_k - 1} \propto \prod_{k=1}^{n} \theta_k^{x_k + \alpha_k - 1} \sim \mathrm{Dir}(x + \alpha)
\end{aligned}
$$

This is because the Dirichlet is a distribution over a vector α of concentration parameters. Hence, when θ is sampled, θ represents a probability distribution over categories, and sampling a x given a θ is equivalent to sampling from a multinomial.

When used as a prior, the symmetry is particularly useful, as there is no prior knowledge favoring one component over another: in this case, all of the elements making up the parameter vector α have the same value, and the distribution can be parametrized by a single scalar value α, called the concentration parameter.

2.2 Dirichlet Process

The Dirichlet process (Ferguson, 1973) provides a framework for modelling non-parametric probability distributions, enabling the sampling of cluster assignments.
Given a measurable space $(\Theta, \mathcal{B})$, the DP is defined by a fixed probability measure G_0 on Θ and a concentration parameter α, the first characterizing the underlying structure of the data, and the second influencing the variability of the process (the higher α, the more variable the process). The Dirichlet Process DP(α_0, G_0) with $\alpha_0 \in \mathbb{R}^+$ is defined as the distribution of a random G over the space of probability measures on Θ, such that, for any finite measurable partition $(A_1, \ldots, A_r)$ of Θ, it is:

$$(G(A_1), \ldots, G(A_r)) \sim \text{Dir}(\alpha_0 G_0(A_1), \ldots, \alpha_0 G_0(A_r))$$

Operatively, the DP takes a Dirichlet distribution G and generates a new distribution based on G, with α_0 regulating how close the new one is to the old one (the higher α, the closer to G). This justifies:

$$\mathbb{E}(\text{DP}(\alpha_0, G_0)) = \mathbb{E}(\text{Dir}(G_0))$$

A way to view the functioning of the process is the so-called *stick breaking construction*, where we iteratively sample $v_i \sim \mathcal{B}(1, \alpha)$ and compute the clusters' weights $\pi_i = v_i \prod_{j=1}^{i-1}(1 - v_j)$. The resemblance to the stick-breaking part can be seen by considering π_i as the length of a piece of a stick: starting with a unit-length stick, at each step we break off a portion of the remaining stick according to v_i, and assign it to π_k. An important remark to point out here relates to the fact that while G_0 may be continuous distribution over Θ, the DP assigns to each atom sampled from G a discrete mass, such that $G = \sum_i \pi_i \theta_{\psi_i}$, with ψ_i i.i.d. from G_0.

When G_0 is continuous, the weights represent the proportions of the total population that belong to each cluster, and the infinite-dimensional nature of the DP allows for an unlimited number of potential clusters. The Dirichlet Process offers a probabilistic mechanism for modeling an infinite mixture model, where the number of mixture components is potentially unbounded and not specified a priori.

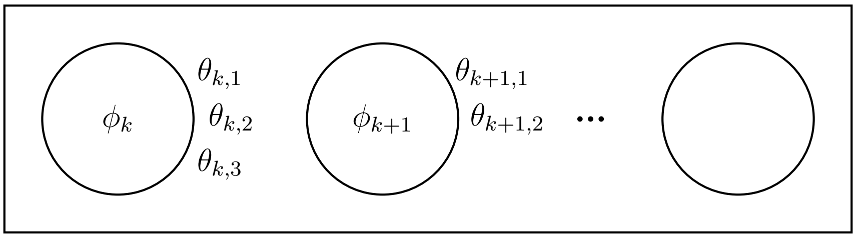

Figure 1: The rectangle is a restaurant and the circles are the tables. The probability of an incoming customer to sit at any table will be proportional to the number of people that are already seated there.

2.3 Chinese Restaurant Process

Consider a restaurant with a potentially unlimited number of tables and clients, ϕ_k indicating the table customer θ_k is assigned to. Whenever a new person comes in, they will sit at an already occupied table with probability proportional to the m_k customers seated there and will choose instead a new table with probability proportional to α_0. Mathematically, we can model the probability for the i^{th} customer, conditioned on the previous ones, to be:

$$\theta_i|\theta_1,\ldots,\theta_{i-1},\alpha_0,G_0 \sim \sum_{k=1}^{\mathrm{K}} \frac{m_k}{i-1+\alpha_0}\delta_{\phi_k} + \frac{\alpha_0}{i-1+\alpha_0}G_0 \quad (1)$$

And it is evident from this that the more frequently a point ϕ_k is drawn, the more it is likely to be drawn in the future. Despite such probability is conditional on the table assignments of previous customers and the tendency to join already occupied tables, the intrinsic randomness of the process makes the probability of sitting at new tables (i.e. generating new clusters, or themes) not negligible. For this reason, it is important to notice that the CRP cannot be used to consistently estimate the number of topics in topic model since it diverges as the sample size grows infinitely.

3 HISTORY OF TOPIC MODELING

Significant progress has been made on the problem of modeling collections of discrete data and different methodologies have evolved through the time, each increasing the level of complexity in attempt to offset the limitations observed in the preceding approaches.
Here I shortly present a list of techniques employed before the formulation of Latent Dirichlet Allocation and Hierarchical Dirichlet Process.

3.1 tf-idf

The *tf-idf* scheme (Salton and McGill, 1983) measures how important a word is to a corpus in a collection, adjusted for the fact that some words appear more frequently in general.
A basic vocabulary of terms is chosen and, for every document, a count of the occurrences of each word is formed. After normalization, this frequency count is multiplied with an inverse document frequency count, which is a proxy for the number of occurrences of a word in the entire corpus, but with lower values for the terms that appear the most. Resulting in a matrix of the form *term-by-document*, with the *tf-idf* values for each of the documents, this scheme reduces any corpus of arbitrary length to fixed-length vectors of real numbers, also enabling the comparison accross collections.

However, while it provides a straightforward mean to identify important terms, this approach is limited in that it does not account for the semantic relationships between words, nor does it capture any underlying structure within the documents, hence it offers little insight into the statistical patterns that may exist within the collection.

3.2 LSI

LSI (Deerwester et al., 1990) introduces an advancement over the previous technique reducing the dimensionality of the term-document matrix: using singular value decomposition, it finds a subspace for the *tf-idf* features that captures most of the variance in the corpus, resulting in significant compression, while preserving important semantic structures and capturing aspects of basic linguistic notions, by projecting terms and documents into a shared latent space.such as synonyms. The *document-by-term* matrix is broken down in the *document-by-topic* and the *topic-by-term* matrices, which measure how topics (terms) are related to documents (topics), whereas the singular values represent each topic's relevance in the corpus.

Unfortunately, the linear nature of dimensionality reduction poses the risk of discarding valuable information particularly in cases where complex, non-linear relationships exist. Moreover, the model struggles with out-of-vocabulary terms that were not present in the train dataset, limiting its flexibility in handling new data.

3.3 Mixture of unigrams

The *Mixture of Unigram* marks a significant shift from the aforementioned algebraic methods as it first formulates a probabilistic framework. Each document is assumed to be generated from a single latent topic z, with the words being sampled from a multinomial distribution conditioned on it. Specifically, the probability of a document is given by:

$$p(\mathbf{w}) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n|z)$$

Where $p(z)$ is the probability of a topic under an empirical distribution and $p(w_i|z)$ is the probability of drawing a word given the topic.

Despite the probabilistic shift, the model is limited by the assumption that each document is generated from a single topic, which is often unrealistic. On top of that, the model's empirical nature does not guarantee an adequate level of flexibility, as it can only produce meaningful results for known documents, i.e. the ones that are part the training set.

3.4 pLSI

The pLSI technique (Hofmann, 1999) builds upon the Mixture of Unigrams and introduces a generative probabilistic model where each document is represented as a different mixture of topics, allowing the words in such documents to be sampled them from a mixture of multinomial random variables, and thus to be different.

Let d be a document label and w_i be a word conditionally independent given an unobserved topic

z. pLSI posits:

$$p(d, w_n) = p(d) \sum_z p(w_n|z)p(z|d)$$

Where $p(d)$ is the probability of a label and $p(w_n|z)p(z|d)$ are the probabilities of choosing a topic and a word given that topic. Each word is generated from a single topic, but different words in the same document may come from different topics, resulting in a more flexible and realistic approach.

Still, the main issue with pLSI is the lack of a predictive approach, which is instead necessary if one wants to integrate unseen documents. In this situation, the Bayesian approach would be an appealing option for inducing predictive rules, but not employing it forces the number of estimated parameters to grow proportionally with the size of the corpus, leading to overfitting and poor generalization.

4 LATENT DIRICHLET ALLOCATION

LDA (Blei et al., 2003) is a generative model for a corpus of texts. Documents are represented as random mixtures over latent topics, where each topic is identified by a distribution over a given set of words.

4.1 The Model

Define a generative process for every document **w** in corpus D:

1. Set the length of the document to N $\sim$ Poisson(ξ)
2. Sample a mixture of topics $\theta \sim$ Dir(α) from a vector α of concentration parameters
3. For each word w_n:
   (a) Choose a topic $z_n|\theta \sim$ Multinomial(θ)
   (b) Choose a word $w_n|z_n, \beta$ with probability $p(w_n|z_n, \beta)$

Some simplifying assumptions are necessary for the model to operate correctly. First of all, the number of topics k in the Dirichlet and the dimension of the matrix parameter β governing the word distributions are fixed. Furthermore, the length N of the document is neglected, as it can be treated independently of other variables and is not crucial to our scope.

Given the parameters α and β, which regulate the mixture of topics and words for the document, respectively, the joint probability of observing a document with topic mixture θ, set of topics **z** and words **w** is:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{2}$$

Where $p(\theta|\alpha)$ samples a mixture of topics in the simplex, $p(z_n|\theta)$ chooses a topic from the mixture and $p(w_n|z_n, \beta)$ selects the words for the document.

It is essential to set LDA apart from other simpler methods, such as the Dirichlet-Multinomial clustering model, as in the latter we have a distribution of topics over the whole corpus, but not over the single documents, implying that each of them will contain terms pertaining to a single topic. Hence, we classify models of this type as *two-level*, while LDA falls into the *three-level* category, providing generally higher flexibility and a better fit for real-world data.

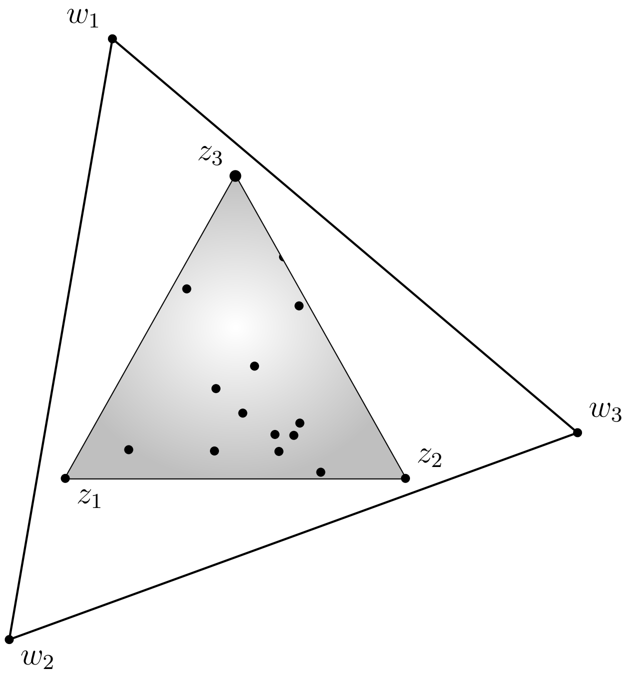

Figure 2: The inner simplex over z is embedded in the outer one over w, the corners of each simplex represent $p = 1$ for a single w_i, or z_i, to come up, and the shading is the distribution over the mixture of topics from which the words are drawn.

Another aspect of LDA is the concept of *exchangeability*. In topic modeling, a relevant issue revolves around the order of the words, but this is often eluded adopting the *bag of words* assumption, stating that the order of the words in a document can be neglected.

To justify this, Blei et al. rely on de Finetti's theorem, which provides a characterization of the invariance with respect to the oredring in the context of conditional independent variables. In our specific case, assuming that the joint distribution is invariant to permutation allows to focus only on the underlying topics that generated the words themselves.

Since in LDA we assume that words are generated by topics under fixed conditional distributions and that those topics are infinitely exchangeable within a document, we can exploit De Finetti's theorem to rewrite (2) as the probability of a sequence of words and topics:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

$$\Rightarrow p(\mathbf{z}, \mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \, d\theta$$

4.2 Comparison with other models

To better understand LDA, it can be useful to compare it with its predecessors, particularly in terms of their geometric interpretations, which helps clarifying the relationship between them. As shown in figure 2, imagine a space, or simplex, representing all possible terms in a document and another representing all possible topics. The first simplex is embedded within the second because it arises, by definition, from combinations of words. For the sake of completeness, we shall point out that each simplex has a governing distribution, with the corners of the outer simplex corresponding to distributions where a single word has probability one to come up. In this setup:

- The *Mixture of Unigrams* samples each word in a document from a single topic chosen among the corners of the topics' simplex
- pLSI draws each word in a training document from a random topic, which is itself chosen under an empirical document-specific distribution over the topics' simplex
- LDA generates the words for both observed and unseen documents from a topic selected by a randomly parameterized distribution over topics.

While it may be difficult to grasp the differences between the approaches from the rest of the discussion, it is evident here that each model is characterized by increasing levels of complexity, gradually improving the ability to capture the intricate patterns of interest.

5 HIERARCHICAL DIRICHLET PROCESS

Having established an understanding of the process in section 2, it is now time to explore its application as a prior distribution in Dirichlet Process Mixture Models and further extend it to the Hierarchical Dirichlet Process (Teh et al., 2004), a non-parametric generative model designed for clustered data, where each group is represented as a mixture of potentially infinite latent topics, and each topic is itself a distribution over a given set of words.

5.1 Dirichlet Process Mixture Model

In a mixture model based on the Dirichlet Process, clusters $\theta_i|G$ are distributed according to a random G and points $x_i|\theta_i$ are drawn independently from a distribution $F(\theta_i)$ over the clusters. Moreover, since G is represented with a stick-breaking construction, θ_i takes on value ϕ_k with probability π_k. If we place now a symmetric Dirichlet prior over a given L number of mixture components, this yields the construction of the following model:

1. Place a prior over the weights $\pi|\alpha_0 \sim \mathrm{Dir}(\alpha_0/\mathrm{L}, \dots, \alpha_0/\mathrm{L})$
2. Sample the mixture components $\phi_k|G_0 \sim G_0$
3. Choose the components from the mixture $z_i|\pi \sim \pi$
4. Draw $x_i|z_i, (\phi_k)_{k=1}^{\mathrm{L}} \sim F(\phi_{z_i})$

So that $G = \sum_{k=1}^{\mathrm{L}} \pi_k \delta_{\phi_k}$. Eventually, we can extend this model to an infinite limit of finite mixture components by letting L $\to \infty$ and this will turn extremely useful for its applicability. Indeed, though with minimum probability, the DP always allows a new cluster to arise.

5.2 The Hierarchical Model

Hierarchical models are characterized by the use of additional layers of hierarchy, increasing the complexity, but also guaranteeing higher flexibility. In our case, HDP is particularly well-suited for grouped data, where each group may exhibit its own unique clustering structure, while still sharing global characteristics across groups, suggesting the need for a level that govern documents assigned to the same topics and another one for documents of the whole corpus.

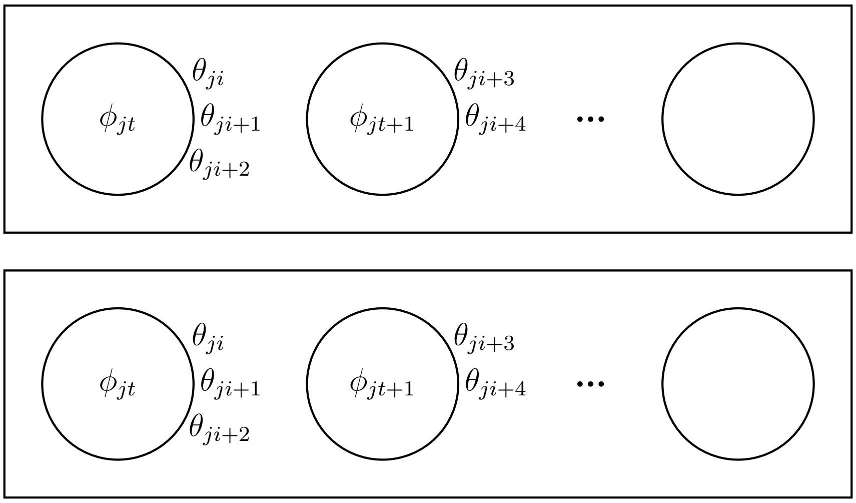

Figure 3: Each rectangle represents a restaurant and the circles are the tables. The probability of an incoming customer to sit at any table will be proportional to the number of people that are already seated there.

To illustrate the concept, recall the Chinese restaurant metaphor and suppose there is now an entire franchise of such restaurants. There is a shared menu (set of parameters) across the whole franchise and a single dish (cluster, or topic, parameters) is served per table (cluster, or topic) as soon as it is populated by the first customers (data, or documents), which implies that all clients seated at the same table will share the same dish, but that different tables, within or across restaurants, may or may not be served with the same dish.

The figure above represents the structure of a restaurant in the franchise, where customer θ_{ji} eats dish ϕ_{jt}, served according to the base measure G_0. Specifically, denote with t_{ij} the table in restaurant j where customer i seats and with k_{jt} the dish served at table t in restaurant j. Moreover, n_{jtk} stands for the number of clients in restaurant j at table t eating dish k, while m_{jk} represents the number of tables in restaurant j serving dish k. Recalling (1) and integrating over the dishes, the probability for a new customer coming into the restaurant is:

$$\theta_{ji}|\theta_{j1},\dots,\theta_{ji-1} \sim \sum_{t=1}^{m_{j.}} \frac{n_{jt.}}{i-1+\alpha_0}\delta_{\psi_{jt}} + \frac{\alpha_0}{i-1+\alpha_0}G_0 \tag{3}$$

If needed, the model provides a way to sample dishes when new tables get occupied. Clearly, there is also the hierarchical analog of the mixture model, where β is a global vector and π_j is already group-specific. Accordingly, we have:

1. Place a prior over the weights $\beta|\gamma \sim \mathrm{Dir}(\gamma/\mathrm{L},\dots,\gamma/\mathrm{L})$
2. Generate the in-group masses $\pi_j|\alpha_0, \beta \sim \mathrm{Dir}(\alpha_0\beta)$

3. Sample the mixture components $\phi_k|G_0 \sim G_0$
4. Choose the components from the mixture $z_{ji}|\pi_j \sim \pi_j$
5. Draw $x_{ji}|z_{ji}, (\phi_k)_{k=1}^{\mathrm{L}} \sim F(\phi_{z_{ji}})$

So that $G_0 = \sum_{k=1}^{\mathrm{L}} \beta_k \delta_{\phi_k}$ and $G_j = \sum_{k=1}^{\mathrm{L}} \pi_{jk} \delta_{\phi_k}$. The stick-breaking condition is respected both at a base and at a group-specific level, indicating that we have induced a DP on an additional layer. This structure ensures that clusters are shared across groups, but the relative importance of each cluster can vary from group to group, allowing for both global consistency and local variability in the data.

The HDP's ability to model complex, hierarchical data structures makes it a powerful tool in many applications, particularly in scenarios where the number of underlying clusters is unknown and potentially infinite. Its flexibility and non-parametric nature allow it to adapt, providing a robust framework for modeling grouped data with shared characteristics.

6 INFERENCE

Inference represents the main problem in topic models, and it is concerned with estimating the latent variables and parameters from the observed data. More precisely, it involves determining the distribution of topic proportions and assignments for each word in a document, conditioned on the observed words and the hyperparameters of the model.
However, this is a very complex task. In LDA, for example, we could try use the probabilistic framework defined in the previous section, aiming to compute the posterior distribution of the latent variables, namely the topic proportions: using (2) and conditioning on the given docs, we would get:

$$\begin{aligned} p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) &= \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \\ &= \frac{p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)}{\int p(\theta|\alpha) \prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta)\, d\theta} \end{aligned}$$

Unfortuately, we notice that such distribution is generally intractable, due to the high-dimensional integral at the denominator and the exponential number of available topic assignments, which results in an extreme computational burden and forces us to switch to alternative approximation techniques. There are several approaches for that, each offering a different trade-off between computational efficiency and accuracy: in this section I will discuss the family of variational inference algorithms and the Gibbs sampling approach as a MCMC counterpart. Starting from the latter, I will delve into the methods, explore how they work, mention their advantages and limitations, and how they can be applied to perform inference effectively.

6.1 Gibbs Sampling

The Gibbs sampling algorithm is a Markov Chain Monte Carlo method that keeps sampling each variable conditioned on the current value of the others, gradually refining the distribution it is meant to estimate. For this reason, it is particularly suitable in frameworks where it is possible to leverage the conditional dependencies among the variables, such as topic models.

The algorithm first draws a random assignment of topics to words, and then it iteratively updates the probability to observe each data point, given all the other variables, upon converging to a stable state. This means that the conditional density of a single point x_{ji} under mixture component z_i and given the data is:

$$\begin{aligned} f_{z_i}^{-x_{ji}}(x_{ji}) &= \frac{f((x_{ji}|\phi_k), x^{-ji})}{f(x^{-ji})} \\ &= \frac{\int f(x_{ji}|\phi_k) f(x^{-ji}|\phi_k) g(\phi_k)\, d\phi_k}{\int f(x_{-ji}|\phi_k) g(\phi_k)\, d\phi_k} \end{aligned}$$

Where the joint probability is equivalent to the product because x_{ji} are i.i.d. from F. Now, we can use this framework to adapt the Chinese restaurant franchise and produce a Gibbs sampling scheme for estimating posterior inference on the latent variables, namely the number of tables and dishes, given the observed data. Exploiting the density calculated above and recalling (3), it is possible to define the conditional distribution of both tables and dishes.

Gibbs sampling is particularly useful in the context of HDP, and an interesting feature of it is that at some point, a table may become unpopulated: in this case, since the probability of new customers to sit there would be null, no one else will ever eat the dish served at that table, thus making it logic to remove the table and dish from the structure. This practice slightly simplifies the somewhat involved bookkeeping procedure of this scheme, especially if compared to others, as the items are first assigned to the tables and then to the mixture components. Nonetheless, the fact the component membership of one whole table can be changed implies that the membership of multiple data can change at the same time, thus improving its performance.

Operatively, Gibbs sampling initially randomize the topic assignment to each word, and slowly updates it, potentially until convergence. This amounts to sample a topic assignment for a word, given all the other parameters, which, in math, is denoted as:

$$p(\mathbf{z}_{d,n} = k|\mathbf{z}_{-d,n}, \mathbf{w}, \alpha, \lambda) = \frac{p(\mathbf{z}_{d,n}, \mathbf{z}_{-d,n}|\mathbf{w}, \alpha, \lambda)}{p(\mathbf{z}_{-d,n}|\mathbf{w}, \alpha, \lambda)}$$

with $p(\mathbf{z}_{d,n} = k)$ being the probability that the topic for word n in document d is exactly z. Then, integrating out the topics and per-document topic proportions yields:

$$p(\mathbf{z}_{d,n} = k|\mathbf{z}_{-d,n}, \mathbf{w}, \alpha, \lambda) = \frac{n_{d,k} + \alpha_k}{\sum_{i=1}^{K} n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \lambda_{w_{d,n}}}{\sum_{i=1} v_{k,i} + \lambda_i} \tag{4}$$

where $n_{d,k}$ counts how many times document d uses topic k, $v_{k,w_{d,n}}$ stands for the number of times topic k has used the word n from topic d, α_k and $\lambda_{w_{d,n}}$ are the Dirichlet parameters for the document-topic and topic-word distribution, respectively, and the denominators act as normalization terms. Essentially, the left factor measures how much the document likes topic k and the right one is a proxy for how much the topic likes the word $w_{d,n}$.

After initial randomization, we go through each word in all the documents of the corpus, remove the current topic assignment, and reassign it according to the above equation, that is, according to how much the document likes the topic and how much the topic likes the word.
We can compute such probability for each of the topic, hence at the end selecting the topic to which the word shall be assigned to is like sampling from a categorical distribution.

It is therefore possible to present an algorithm to fit an LDA model on a collection of documents using a Gibbs sampling approach:

```
while not converged do
    for $w_{d,n}$ assigned to $z_x$ do
        $n_{d,z_x} = n_{d,z_x} - 1$
        $v_{z_x,w_{d,n}} = v_{z_x,w_{d,n}} - 1$
        set $z_{x'} = k \propto \frac{n_{d,k}+\alpha_k}{\sum_{i=1}^{K} n_{d,i}+\alpha_i} \frac{v_{k,w_{d,n}}+\lambda_{w_{d,n}}}{\sum_{i=1} v_{k,i}+\lambda_i}$
        $n_{d,z_{x'}} = n_{d,z_{x'}} + 1$
        $v_{z_{x'},w_{d,n}} = v_{z_{x'},w_{d,n}} + 1$
```

In the HDP framework, the number of topics in not fixed, thus requiring relevant modification to adapt the above sampling algorithm. Since it's possible to either reassign a word to an existing topic or to a new topic, the algorithm should not only consider the likelihood of associating the word to one of the existing topics, but it should also include the possibility of creating a new topic by incorporating the stick-breaking process.
Moreover, the hierarchical structure poses some complications in the update phase. Firstly, the Gibbs sampler must account for the possibility of potential variations in the document-specific distribution by updating also the global and local topic distributions. Secondly, the equations for the topic assignments update in HDP are more complex, because the sampler also needs to consider global topic counts and hierarchical dependencies, with the probability of assigning a word to a topic now including terms from both the document-specific and global DP, which require careful bookkeeping during the sampling process. With these careful considerations, it becomes possible to adapt the algorithm presented for the LDA model to the HDP case.

Still, there are some intrinsic limitations to the applicability of this framework, for example, a couple of assumptions are necessary. The first relates to the base measure G_0, which must be conjugate to the distribution F: in fact, although the non-conjugate case is manageable, posing this removes issues related to approximations and let us focus on our main objective. Also, it is convenient to fix α_0 and γ a priori: there are ways to sample them, but we do not do it for the sake of simplicity. For these reasons, this method can be viewed as a starting point for developing better inference procedures, such as the *collapsed* version.

6.1.1 Collapsed Gibbs Sampling

Collapsed Gibbs Sampling is a variant of the standard Gibbs Sampling algorithm that operates in a *collapsed* space, where the model parameters are marginalized out, and the latent variables are the only ones actually sampled.

In the standard algorithm, the scheme requires sampling the model parameters and the latent variables at the same time, but this can lead to slow convergence, especially when the two are strongly coupled. The issue arises because the sampling process must navigate a space where parameters and latent variables interact in complex ways, hence marginalizing out the first would reduce the dimensional space solely around the latent variables, which may lead to more efficient exploration of the posterior distribution and, consequently, faster convergence.

Indeed, integrating out the model parameters brings several benefits, among which it is worth to mention the reduced computational complexity due a lower number of parameters to compute at each iteration. Moreover, as anticipated, the dependencies between latent variables are reduced, thus making it easier for the algorithm to make his way through the sampling space. Overall, collapsed Gibbs Sampling can produce more accurate estimates of the posterior distribution, as the marginalization process averages over all possible parameter configurations, making the posterior estimates less sensitive to the initial parameter settings and more reflective of the true underlying structure of the data.

It is also interesting to anticipate that the improved mixing properties of a collapsed sampler will stand as motivations behind the development of the *collapsed* version for the variational inference algorithm, which we will see in the next section.

6.2 Variational Inference

The family of variational inference algorithms transforms the problem of posterior estimation into an optimization problem: rather than directly computing an intractable posterior, it aims to approximate it by finding a member of a tractable family of distributions that is closest to the true posterior, in terms of a specific divergence measure. To do so, we use a family of densities over the latent variables, parameterized by a set of free variational parameters, and we find the setting of the parameters that makes our distribution closest to the conditional of interest, so that we can use it as a proxy for the exact conditional density.

First, we posit a family of approximate densities over the latent variables, then, we seek for the member of that family that minimizes the Kullback-Leibler (KL) divergence from the exact posterior, where this KL divergence quantifies the divergence between the approximate distribution and is defined as:

$$\text{KL}(q(\mathbf{z}), p(\mathbf{z}|\mathbf{x})) = \mathbb{E}\left[\log q(\mathbf{z})\right] - \mathbb{E}\left[\log p(\mathbf{z}|\mathbf{x})\right]$$

However, directly minimizing this KL divergence is often not feasible because the formulation relies again on the intractable distribution over the observed variables. To avoid this, variational inference shifts the objective of the optimization: instead of minimizing a distance, we aim to maximize the so called *evidence lower bound*, a surrogate objective defined as:

$$\text{ELBO}(q) = \mathbb{E}_q\left[\log p(\mathbf{z}, \mathbf{x})\right] - \mathbb{E}_q\left[\log q(\mathbf{z})\right] \tag{5}$$

Which we can derive as a lower bound for $\log p(x)$ using Jensen's inequality and rewrite as follows:

$$\begin{aligned}
\log p(x) &= \log \int p(\mathbf{z}, \mathbf{x}) \frac{q(\mathbf{z})}{q(\mathbf{z})}\, d\mathbf{z} \\
&= \log \mathbb{E}_q\left[\frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})}\right] \\
&\geq \mathbb{E}_q\left[\log \frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})}\right] \\
\Rightarrow \text{ELBO}(q) &= \mathbb{E}_q\left[\log p(\mathbf{z}, \mathbf{x})\right] - \mathbb{E}_q\left[\log q(\mathbf{z})\right] \\
&= \mathbb{E}_q\left[\log p(\mathbf{z})\right] + \mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathbb{E}_q\left[\log q(\mathbf{z})\right] \\
&= \mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})\right] - \text{KL}(q(\mathbf{z}), p(\mathbf{z}))
\end{aligned}$$

Maximizing the ELBO brings the approximate distribution closer to the desired posterior and enhances the model's ability to explain the observed data. As a matter of fact, it can be rewritten to emphasize its two components: $\mathbb{E}_q\left[\log p(\mathbf{x}|\mathbf{z})\right]$ is an expected log likelyhood that encourages densities placing their mass on the configurations of latent variables that best explain observed data, while $(q(\mathbf{z}), p(\mathbf{z}))$ is the KL divergence encouraging the choice of an approximate posterior distribution that is close to the prior.

In practice, a common choice for the variational distribution $q(\mathbf{z})$ is the mean-field family, where the latent variables are assumed to be mutually independent and the joint distribution factorizes:

$$q(\mathbf{z}) = \prod_{i=1}^{n} q_i(z_i)$$

Moving to our specific inference problem, it is logic to choose the following variational distribution:

$$q(\theta, \mathbf{z}|\gamma, \varphi) = q(\theta|\gamma) \prod_{i=1}^{n} q(z_i|\varphi_i) \tag{6}$$

Where $q(\theta|\gamma)$ is the Dirichlet over the mixture of topics and $q(z_i|\varphi_i)$ is the probability of topic z_i. Instead, γ and φ are the variational parameters for the distribution governing the latent variables, which do not include the words w, as they are the observed features.

Now that we have the variational distribution, we try to approximate the true objective function, that is $p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$, by maximizing ELBO$(q)$. Recalling (5) and exploiting the independence of the latent variables in (6), we can derive:

$$\begin{aligned}
\text{ELBO}(q) &= \mathbb{E}_q\left[\log p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)\right] - \mathbb{E}_q\left[\log q(\theta, \mathbf{z}|\gamma, \varphi)\right] \\
&= \mathbb{E}_q\left[\log p(\theta|\alpha)\right] + \mathbb{E}_q\left[\log p(\mathbf{z}|\theta)\right] + \mathbb{E}_q\left[\log p(\mathbf{w}|\mathbf{z}, \beta)\right] + \\
&\quad - \mathbb{E}_q\left[\log q(\theta|\gamma)\right] - \mathbb{E}_q\left[\log p(\mathbf{z}|\varphi)\right]
\end{aligned}$$

As previously anticipated, the derived posterior γ and φ correspond exactly to a Dirichlet and multinomial, respectively. Because the updates for the variational parameters can be computed in closed-form, we may write the optimization procedure with the following algorithm:

There are a couple of things to notice here. For example, proper initialization is crucial: rather than starting with a uniform initialization of the parameters, it is fundamental not to initialize the parameters uniformly, but to exploit some information provided by a small subset of documents to guide the optimization in a more probably-right direction from the beginning. In terms of

```
initialize $\varphi_{ij}^0 = 1/k$
initialize $\gamma_j = \alpha_j + \mathrm{N}/k$
while not converged do
    for $i = 1 \to n$ do
        for $j = 1 \to k$ do
            $\varphi_{ij}^{t+1} = \beta_{jW_i} \exp(\psi(\gamma_j^t))$
        normalize $\varphi_i$
    $\gamma^{t+1} = \alpha + \sum_{i=1}^{N} \varphi_i^{t+1}$
```

complexity instead, we can see that the complexity of each step is $\mathrm{O}((\mathrm{N}+1)k)$, because there's the need to evaluate all topics for each word, plus normalization.

Once the parameters have been updated, the actual parameter estimation is performed employing a simple *expectation-maximization* (EM) procedure, essential to maximize the lower bound for the true parameters of the distribution. In the E-step, the optimal values for γ, φ are computed according to the algorithm described previously, while in the M-step, the maximum likelyhood for each document is estimated under the previously approximated variational posterior.

A relevant problem that arises in large text corpora is the sparsity of the vocabulary, which makes it highly likely to find in the test set terms not previously encountered in the train set. This is critical, because if the model has never learnt a specific word during training, such word will come up in a new generated document from the model with null probability. To solve this, Blei et al. adopt a simple smoothing technique, including a Dirichlet smoothing on the multinomial parameter that regulates the distribution over the words.

However, making β conditioned on a new parameters implies treating it as an additional latent variable to estimate using the observable data. For this reason, we can reformulate a more complete variational posterior inference as:

$$q(\beta, \theta, \mathbf{z}|\lambda, \gamma, \varphi) = \prod_{j=1}^{k} \mathrm{Dir}(\beta_j|\lambda_j) \prod_{d=1}^{m} q(\theta_d|\gamma_d) q(\mathbf{z}_d|\varphi_d)$$

Where the rightmost distribution is now explicited at a document level. This formulation requires also a document-level distribution, leading to the optimization of the new variational parameter in the algorithm described previously:

$$\lambda_{ij} = \eta + \sum_{d=1}^{m} \sum_{i=1}^{n} \varphi_{dij} w_{di}^{j}$$

Though this adds a layer of complexity, the model is still tractable and the only two hyperparameters remain α and β. It is astonishing how, with such a slight adjustment, it is possible to sidestep a significant problem, mitigating the issue of unseen words in new documents.

In summary, variational inference is a scalable and efficient method for approximate inference in topic models. Given the closed-form optimization for the parameters, it converges relatively quickly, but it can be computationally demanding compared to methods like Gibbs sampling. Nonetheless, its ability to handle large datasets and provide good approximations makes it a popular choice in practice.

6.3 Stochastic variational inference

Stochastic Variational Inference (SVI) is an extension to the traditional algorithm, that employs stochastic optimization techniques to reduce computational costs and enhance the scalability: unlike batch variational inference, which requires processing the entire dataset before updating the global parameters, SVI does the update by sampling one document at a time.

Before diving into SVI, it is essential to make a punctualization and introduce a new concept. The first is just a matter of terminology, namely the distinction between local and global variables in a model. With *local* variables, we refer to the variables that are specific to individual documents: they capture the latent structure that varies accross documents, that is, the topic assignments and the document-topic distributions. The *global* variables are instead shared across all documents in the corpus and capture the underlying structure that is common across the entire dataset, such as the topic-word distribution.

For what concerns the concept to be introduced instead, it is an alternative to the well-known gradient: the natural gradient. The gradient we are all used to is, by construction, related to the Euclidean geometry, which does a poor job when it comes to measure the divergence between probability distributions. For example, The distributions $\mathcal{N}(0, 10000)$ and $\mathcal{N}(10, 10000)$ are almost indistinguishable, and the Euclidean distance between their parameter vectors is 10. In contrast, the distributions $\mathcal{N}(0, 0.01)$ and $\mathcal{N}(0.1, 0.01)$ barely overlap, but this is not reflected in the Euclidean distance, which is only 0.1. To correct this, we first redefine a natural measure of dissimilarity between probability distributions as the symmetrized KL divergence:

$$D(\lambda, \lambda') = \mathbf{E}_\lambda[\log \frac{q(\beta|\lambda)}{q(\beta|\lambda')} + \mathbf{E}'_\lambda[\log \frac{q(\beta|\lambda')}{q(\beta|\lambda)}]$$

This metric is invariant to parameter transformation, thus being dependent on the distributions only and solving the aforementioned problem. Now, the direction of steepest ascent is exactly the direction of the natural gradient, that is, the direction of steepest ascent in the Riemannian space, where local distances are defined by KL divergence rather than by the L_2 norm. Applying this new framework to our old ELBO, it becomes possible to compute the natural gradient with respect to the variational parameters.

The inefficiency of the standard variational inference algorithm lies in the update for the topic parameter λ described in equation (1), which requires summing over all variational parameters for every word in the collection. In this sense, a slight improvement to the algorithm is already brought with the batch version for variational inference, where the local parameters are updated through coodrinate ascent and the variational parameters λ requires to process all documents before performing the update, but only a batch, i.e. a subset of documents are examined. Still, batch variational inference is inefficient for large collection of documents: computing the local variational parameters multiple times is wasteful, especially at the beginning of the algorithm, when the topics have just been initialized randomly.

Once we understand how batch variational inference works, moving to stochastic variational inference is relatively easy. As a matter of fact, the latter follows the same routine of the former, alternating the updates of local and global parameters, but, at each iteration, a single document is evaluated: this is why it's called *stochastic*, in the same spirit of the sthocastic gradient descent, where a single data point is evaluated before updating the parameters instead of passing through the whole training set. This requires some adjustments in the subsequent iterations, but it is provably faster and still consistent.

The structure of the SVI algorithm is similar to standard variational inference, but there are some slight variation. For instance, the update for ψ is different because it is the expectation of the mean filed variational inference with the natural gradient instead of the euclidean one. Moreover, the computation of an intermediate lambda value is needed to bridge the gap between local and global variables, as in the local updates only a single document is considered, and we want to limit stochastic updates.

```
initialize $\lambda$ randomly
set the step size $\rho_t$ appropriately
while not converged do
    sample a document $w_d$ from the corpus
    initialize $\gamma_{dj} = 1$ for $j \in \{1, ..., K\}$
    while not converged do
        for $i \in \{1, ..., N\}$ do
            set $\psi_{dij} \propto \exp(\mathbb{E}[\log \theta_{dj}] + \mathbb{E}[\beta_{j,w_{di}}])$
        $\gamma_d = \alpha + \sum_{i=1}^{N} \varphi_{di}$
    for $j \in \{1, ..., K\}$ do
        set $\hat{\lambda}_j = \eta + D \sum_{i=1}^{N} \psi_{dij} w_{di}$
    set $\lambda^{t+1} = (1 - \rho_t)\lambda^t + \rho_t \hat{\lambda}$
```

The main difference with respect to batch variational inference lies in the update of the global variational parameter: in the first case, we must go through the whole batch before performing the global parameter update, yielding

$$\hat{\lambda}_j = \eta + \sum_{d=1}^{D} \sum_{i=1}^{N} \psi_{dij} w_{di}$$

whereas in the second, sampling a single document is equivalent to posing that the whole corpus is equal to the document itself, thus making it possible to remove the sum and simply multiply times the number of documents in the collection. Finally, notice how the ultimate update for the lambda recalls the update of a gradient descent: this is exactly the motivation for the algorithm, that of descending down the gradient, which in this specific case is the natural gradient.
In a very similar way, we can follow the local-global routine update in the case of HDP model, with the global parameters now including not only the topic-word distributions, but also the corpus-level breaking proportions.

As proven by the experiments whose results are displayed in figure 4, stochastic variational inference provides some advantages with respect to standard variational inference. Not using exact information, because it's too complicated, but rather a stochastic version of it, and updating the global parameters incrementally, reduces the need for extensive computational resources and speeds up the training process. In the same direction, by leveraging the natural gradient and

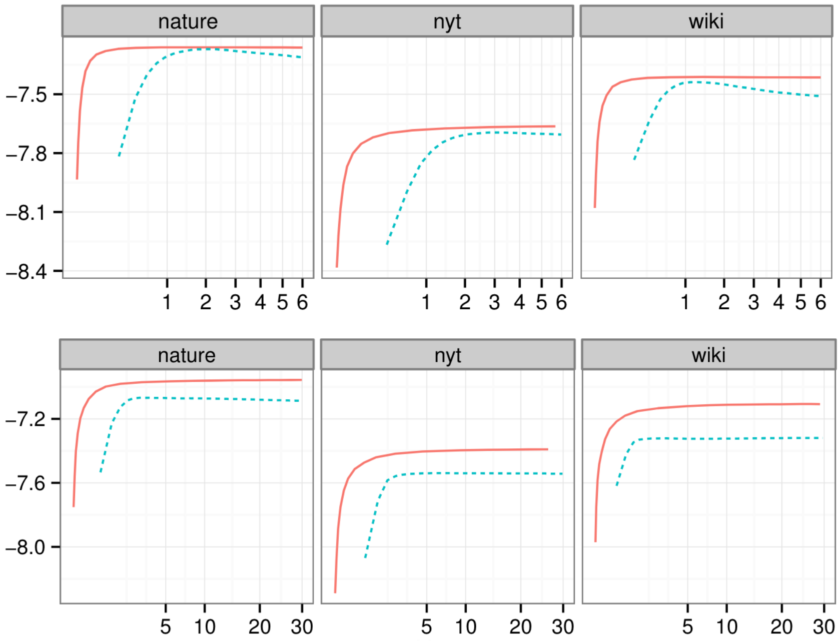

Figure 4: The charts show the convergence of SVI (orange) and Batch Inference algorithms (blue) in terms of negative log perplexity over number of hours. Both with LDA (above) and HDP (below), SVI converges faster and does a slightly better job at modeling the corpus.

stochastic updates, SVI converges more rapidly than batch variational inference, with an evident gap especially in the early iterations. Lastly, the use of the natural gradient solves the problem of the in-adaptability of Euclidean distance to parameter transformation and makes SVI more robust.

6.4 Collapsed Variational Inference

Collapsed variational inference leverages the insight that a sampler operating in a collapsed space where the parameters are marginalized out mixes better than a Gibbs sampler handling both parameters and latent topic variables simultaneously. Similarly, this technique focuses on directly approximating the posterior distribution of the latent variables, rather than simultaneously approximating parameters and latent variables.

On one hand, this suggests that parameters and latent variables are intimately coupled, but, on the other, it is true that the dependencies between latent variables induced by marginalizing out the parameters is expected to be small: the latent variables are assumed to be independent if conditioned on the parameters. This suggests that the mean field, fully factorized approximation assumptions are better satisfied in the collapsed space of latent variables than in the joint space of latent variables and parameters.

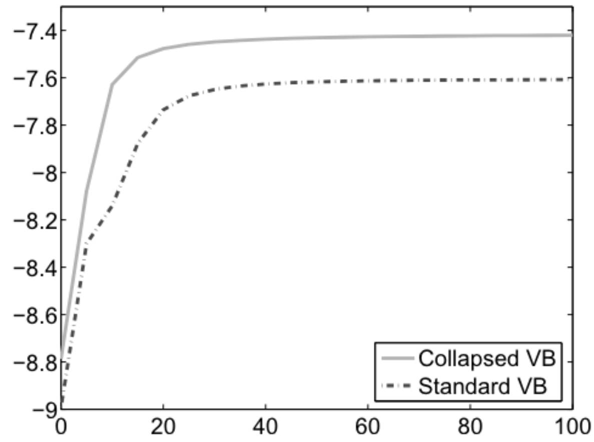

Figure 5: Per-word probability of standard and collapsed variational inference on the NIPS data. In the first iterations, the two models do a similar job at modelling the corpus, but the collapsed version outperforms the standard one in the long run.

One of the challenges of collapsed variational inference is the need to compute expectations over the collapsed space, and this can be computationally demanding: specifically, the process involves calculating Bernoulli averages, which are complex. However, they can be approximated via Gaussian distribution, increasing the algorithm's computational efficiency while preserving its accuracy. To sum up, Collapsed Variational Inference represents a significant improvement by operating in a space where the dependencies among latent variables are minimized. As in the context of Gibbs sampling, the reduced dimensionality of the problem leads to faster convergence and improved accuracy, as shown in figure 5.

7 APPLICATIONS

The models presented here are powerful tools for understanding and organizing large collections of unstructured text data: they have broad applicability, especially in the context of document modeling and classification.
Both LDA and HDP are able to model documents as mixtures of topics, where each topic is a distribution over words. The generative nature provides a rich representation of the document corpus, capturing the underlying thematic patterns, whereas the probabilistic nature allows learning from large corpora without prior knowledge of the document content.

In the specific context of document classification, a significant challenge is feature selection, namely spotting the most relevant aspects of a text to accurately categorize documents. In more traditional approaches, individual words are used as features, leading to a high-dimensional spaces, which make the classification task computationally expensive and prone to overfitting. Topic models offer a solution to this problem by performing dimensionality reduction: the posterior distribution of the Dirichlet associated with each document, representing the document's mixture of topics, can be used a subset of some pre-defined factors in a feature vector. This approach not only reduces the number of features but also captures the semantic structure of the documents. Moreover, the classifier can leverage the thematic information coming from the fact that documents are now represented in terms of their topic distribution, which is more robust to noise than individual word frequencies.

However, while dimensionality reduction can lead to more efficient classification models, it also poses the risk of sacrificing crucial pieces of information. Indeed, the topics identified are just summaries of the document content, and important details for distinguishing between closely related categories might be lost. Luckily, empirical evidence from various studies shows that this does not seem threatening the model's performance, since they still display significant improvements in document classification accuracy, which remains our main objective.

Talking about the HDP model specifically, it is worth to reiterate that it introduces the Dirichlet Process to model the uncertainty related to the total number of topics, which is not fixed. The document-specific mixing proportions requires one DP per document and HDP further tries to endow different DPs with some shared mixture components. Eventually, it is possible to extend it to model multiple corpora, including an additional level of hierarchy in the Process. Now, the first DP is accountable for the corups of corpora, the second generates the base measure for the specific corpus and the third is responsible for the single documents: since they all come from the same top-level DP, topics are shared not only within, but also across corpora.

7.1 Experiments

In this section it is described an experiment conducted to demonstrate the functioning of the variational algorithm for posterior parameter estimation and the difference between the parametric and the non-parametric approach. Since in this thesis I'm discussing topic modeling, it was interesting to test the two models on a corpus of documents, and see how well they perform in recognizing topics.
For these reasons, I decideded to use a set of articles from the proceedings of the Neural Information Processing Systems (NIPS), which I found unprocessed here. The NIPS conference deals with a range of topics covering both human and machine intelligence, and the dataset contains thousands of documents, which made the fitting and testing processes quite slow. However, I chose this classic dataset due to its relevance in the context of language processing and because it is categorized into nine major sections, suggesting that some topics could be shared across documents.

To avoid the models recognize topics around the most frequent words, which would lead to higher accuracy but less meaningful results, I adopted some standard preprocessing techniques in the literature, such as tokenization and lemmatization. After that, I removed common stop-words and discarded extremely low or extremely high importance words, appearing in less than 10%, or more than 90%, of the documents. This step ensures that the topics caught reflect the actual content of the documents rather than superficial word frequency patterns.
Finally, the models were evaluated through 5-fold cross-validation and the metric used was the perplexity, defined as the exponential of the negative average log likelihood:

$$exp(-\frac{1}{n}p(\mathbf{w}|\text{corpus})) \tag{7}$$

with n being the total number of words. The likelihood of a word is computed as the sum of its

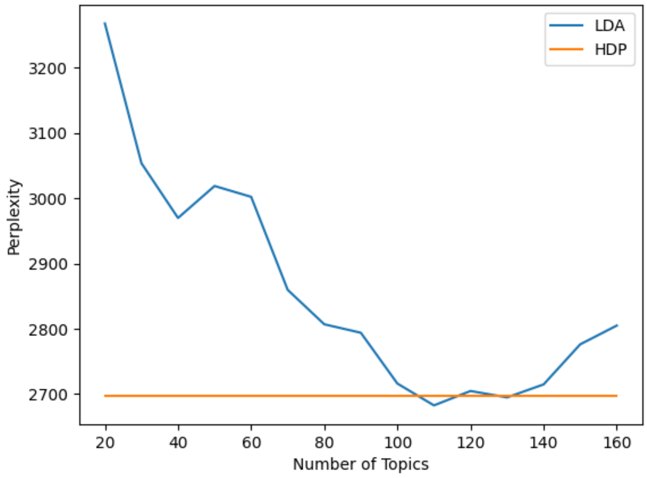

Figure 6: Comparison of LDA and HDP perplexities over the train corpus. While different LDA models are needed to test different number of topics, a single HDP is trained.

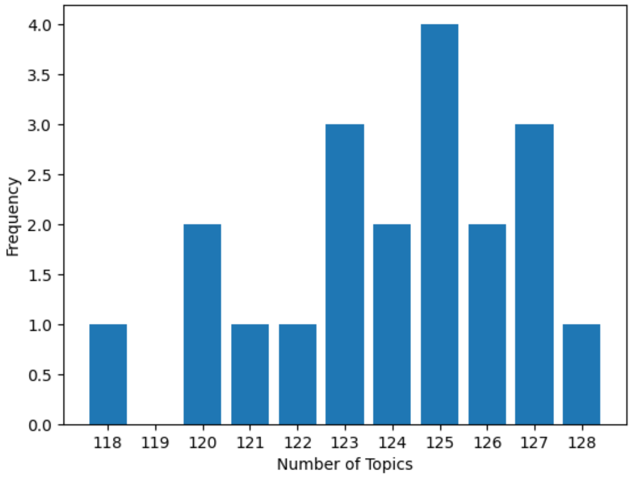

Figure 7: Number of topics inferred over the train corpus sampled by the posterior distribution fit with the HDP model.

probability to appear in any topic, discounted by the mixture value of that topic to actually be a theme in the document, and the total likelihood of the document is the summed likelihood of all words.

The two models were kept as similar as possible, using symmetric Dirichlet distributions for both the topics-per-document and words-per-topic priors, same chunk size and maximum number of iterations. I used the Gensim library, which is an easy-to-use open-source tool implementing main models and other useful functions in the preprocessing steps. The implementations are directly based on the papers that first demonstrated the use of variational inference for online learning in the context of LDA and HDP, respectively.

The results of the experiment are shown in Figures 2 and 3: the perplexity for multiple LDA models, with the number of topics varying from 20 to 160, was evaluated and compared with that from a single HDP model. Interestingly, while the best LDA models slightly outperformed the HDP model in terms of perplexity, the trade-off favored HDP in terms of time spent on parameter tuning. Unlike LDA, which requires careful tuning of the number of topics, HDP automatically infers the number of topics from the data, making it more convenient in practical applications.

It is worth to mention that, on average, the number of topics sampled by the HDP was slightly higher than the optimal number of topics for the LDA model, signaling a slight difference between the two models. Moreover, notice how LDA is extremely poor at modeling the corpus when trained with a largely incorrect number of topics, highlighting once again the remarkable gap between the two models in terms of flexibility.

The objective of this analysis, inspired from the one conducted by Teh et al. in their paper, was to compare the performance of one the algorithms presented in this thesis on two prominent models for topic modeling. Future work could include the exploration of other inference algorithms, such as Gibbs Sampling or variations on the VB theme.

8 DISCUSSION

8.1 Models

It should be clear at this point that the models described here represent important advancements in the field of topic modeling, but it is also necessary to consider their difference and limitations, which make them not optimal, or at least improvable in some aspects.

Latent Dirichlet Allocation is a generative probabilistic model that is widely used due to its simplicity, ease of implementation, and well-established inference techniques. One of the main strengths of LDA is its ability to produce interpretable topics, useful in various domains, but its requirement to pre-specify the number of topics can sometimes be problematic. LDA struggles when the number of topics is relatively low or extremely large, i.e. with an excessively small or large corpus. On one side, it may lack some information to infer the underlying topics accurately, while, on the other, processing large corpora can be computationally expensive, especially when dealing with a vast number of documents and a large vocabulary. Moreover, it relies on the bag of words assumption and disregarding the word order within the document may prevent capturing the semantic relationships between words either in a single sentence or in the whole document.

One of the improvements HDP offers is that it solves the problem related to the fixed number of topics, but this higher degree of flexibility comes with different prior distributions and additional layers in the model, thus making it computationally expensive and slower, especially for larger datasets. On top of that, this method relies on specific hyperparameters, like the concentration for the Dirichlet Process. These can significantly influence the performance of the model, and selecting appropriate values requires either knowledge or experimentation, which opposes the need to limit the number of trials given the complexity of the calculations.

Finally, an issue concerning both approaches is that of the interpretability of topics. In fact, the models return a set of topics and a list of words per topic, but the job of inferring a name for the topic is up to the developer and it may not always be straightforward, especially as the number of potential topics grows.

In summary, LDA is a powerful tool when the number of topics is known and remains consistent across the dataset: it is easier to implement and interpret, making it suitable for a wide range of applications where computational efficiency and interpretability are key bindings. HDP instead offers a more flexible and adaptive approach, automatically inferring the number of topics, which is particularly useful for complex and large-scale datasets. However, this flexibility costs in increased computational complexity and potentially challenging model interpretation.

8.2 Inference algorithms

Each of the algorithms discussed in the inference section offer distinct advantages and trade-offs depending on the specific application and requirements of the model.

Starting from the Gibbs Sampling scheme, it is a straightforward MCMC method with notable simplicity and ease of implementation. However, it often suffers from slow convergence, especially in high-dimensional spaces or when dealing with highly correlated variables. Moreover, it can be computationally expensive, as it typically requires a large number of iterations to achieve accurate results. In this sense, the collapsed version reduces the problem dimensionality, leading to faster convergence and better mixing properties, with more accurate approximations.
Still, the implementation is somewhat more complex due to the need to derive and compute the marginalized distributions, and the fact that not all models or variables can be easily collapsed limits the applicability of this method.

In contrast, Variational inference is a deterministic approach known for its scalability and speed, making it particularly suitable for large datasets. However, the quality of the approximation strongly depends on the chosen variational family, which might not always capture the true posterior accurately, particularly in complex models, where it may be trapped in local minima.

Stochastic VI builds on the standard version, incorporating stochastic optimization techniques that allows to reduce computational burden and scale up to large datasets, also enabling online learning, where the model is updated as new data arrives. Although it introduces significant computational advantages, it inherits some limitations, including the sensitivity to the choice of variational family and potential issues with local minima. Moreover, the stochastic nature of the algorithm introduces variance in the estimates.

Collapsed Variational Inference finally combines the strengths of VI and collapsed Gibbs sampling, with the marginalization yielding simpler optimization and faster convergence. However, it inherits the complications of the collapsed Gibbs sampling, such as a complex implementation requiring the derivation of collapsed variational objectives and an extra computational overhead due to the collapsing step. Again, its applicability is limited by the feasibility of collapsing variables in the given model.

In summary, the choice of the inference algorithm should be guided by the application needs, including size of the dataset, model complexity, and balance between approximation accuracy and computational efficiency. On one hand, Gibbs Sampling offers flexibility and accuracy but may be computationally prohibitive for large or complex models. On the other hand, VI provides scalability and speed, making them suitable for large-scale applications, although they might compromise on the precision of the posterior approximation. Collapsed VI offers a potential middle ground, improving in accuracy at the cost of increased implementation complexity.

References

[1] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. (2009) *On Smoothing and Inference for Topic Models.*

[2] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. (2017) *Automatic Differentiation Variational Inference.* Journal of Machine Learning Research

[3] C. Wang, J. Paisley, D. M. Blei. (2011) *Online Variational Inference for the Hierarchical Dirichlet Process.* Proceedings of the Machine Learning Research

[4] D. M. Blei, A. Kucukelbir and J. McAuliffe. (2017) *Variational Inference: A Review for Statisticians.* Journal of the American Statistical Association

[5] D. M. Blei, A. Ng and M. I. Jordan. (2003) *Latent Dirichlet Allocation*. Journal of Machine Learning Research

[6] G.Salton and M. McGill. (1983) *Introduction to Modern Information Retrieval.*

[7] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. (2013) *Stochastic Variational Inference.* Journal of Machine Learning Research

[8] M. D. Hoffman, D. M. Blei, and F. Bach. (2010) *Online Learning for Latent Dirichlet Allocation.* Neural Information Processing Systems papers

[9] S. Das, Y. Niu, Y. Ni, B. K. Mallick, and D. Pati (2023) *Blocked Gibbs Sampler for Hierarchical Dirichlet Process.* Journal of Computational and Graphical Statistics

[10] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. (1990) *Indexing by latent semantic analysis.* Journal of the American Society of Information Science

[11] T. Hofmann. (1999) *Probabilistic latent semantic indexing*. Proceedings of Twenty-Second Annual International SIGIR Conference

[12] T. L. Griffiths and M. Steyvers. (2004) *Finding Scientific Topics.*

[13] Y. W. Teh., D. Newman, and M. Welling. (2006) *A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation.* Neural Information Processing Systems papers

[14] Y. W. Teh, K. Kurihara, and M. Welling. (2007) *Collapsed Variational Inference for HDP.* Neural Information Processing Systems papers

[15] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. (2006) *Hierarchical Dirichlet Processes.* Journal of the American Statistical Association