UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

THIAGO FREDES RODRIGUES

A Probabilistic and Incremental Model for Online Classification of Documents: DV-INBC

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Paulo Martins Engel

Porto Alegre
April 2016

CIP – CATALOGING-IN-PUBLICATION

Rodrigues, Thiago Fredes

A Probabilistic and Incremental Model for Online Classification of Documents: DV-INBC / Thiago Fredes Rodrigues. – Porto Alegre: PPGC da UFRGS, 2016.

68 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Advisor: Paulo Martins Engel.

1. Topic modeling. 2. Document classification. 3. Incremental learning. 4. Online learning. I. Engel, Paulo Martins. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb
Coordenador do PPGC: Prof. Luigi Carro
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“There is only one corner of the universe you can be certain of improving,
and that’s your own self.”
— ALDOUS HUXLEY

ACKNOWLEDGEMENTS

First I would like to thank my advisor, Dr. Paulo Martins Engel, for his guidance, patience and valuable discussions. I would also like to thank all the people of LIAC (Connectionist Artificial Inteligence Laboratory), for being so helpful and providing smart and useful talks. Specially, I would like to thank my colleagues Renato Pereira and Edigleison Carvalho with whom I shared most of my time at LIAC. I also thank the finantial support provided by CNPq and mention the stimulating research environment found in the Informatics Institute.

I also thank all my colleagues and friends I made in the Institute for providing rich discussions about research and life itself. Fynally, I would like to thank my parents for their neverending support and love. And also for understanding all the times I could not be with them.

RESUMO

Recentemente, houve um aumento rápido na criação e disponibilidade de repositórios de dados, o que foi percebido nas áreas de Mineração de Dados e Aprendizagem de Máquina. Este fato deve-se principalmente à rápida criação de tais dados em redes sociais. Uma grande parte destes dados é feita de texto, e a informação armazenada neles pode descrever desde perfis de usuários a temas comuns em documentos como política, esportes e ciência, informação bastante útil para várias aplicações. Como muitos destes dados são criados em fluxos, é desejável a criação de algoritmos com capacidade de atuar em grande escala e também de forma on-line, já que tarefas como organização e exploração de grandes coleções de dados seriam beneficiadas por eles. Nesta dissertação um modelo probabilístico, on-line e incremental é apresentado, como um esforço em resolver o problema apresentado. O algoritmo possui o nome DV-INBC e é uma extensão ao algoritmo INBC. As duas principais características do DV-INBC são: a necessidade de apenas uma iteração pelos dados de treino para criar um modelo que os represente; não é necessário saber o vocabulário dos dados a priori. Logo, pouco conhecimento sobre o fluxo de dados é necessário. Para avaliar a performance do algoritmo, são apresentados testes usando datasets populares.

Palavras-chave: Topic Modeling, Classificação de Documentos, Aprendizado Incremental, Aprendizado On-line.

ABSTRACT

Recently the fields of Data Mining and Machine Learning have seen a rapid increase in the creation and availability of data repositories. This is mainly due to its rapid creation in social networks. Also, a large part of those data is made of text documents. The information stored in such texts can range from a description of a user profile to common textual topics such as politics, sports and science, information very useful for many applications. Besides, since many of this data are created in streams, scalable and on-line algorithms are desired, because tasks like organization and exploration of large document collections would be benefited by them. In this thesis an incremental, on-line and probabilistic model for document classification is presented, as an effort of tackling this problem. The algorithm is called DV-INBC and is an extension to the INBC algorithm. The two main characteristics of DV-INBC are: only a single scan over the data is necessary to create a model of it; the data vocabulary need not to be known a priori. Therefore, little knowledge about the data stream is needed. To assess its performance, tests using well known datasets are presented.

Keywords: Topic modeling. document classification. incremental learning. online learning.

LIST OF ABBREVIATIONS AND ACRONYMS

INBC	Incremental Naive Bayes Clustering
DV-INBC	Dynamic Vocabulary INBC
LDA	Latent Dirichlet Alocation
SVD	Singular Value Decomposition
PLSI	Probabilistic Latent Semanting Indexing
LSI	Latent Semantic Indexing
BOW	Bag of words
EM	Expectation-Maximization
RBM	Restricted Boltzman Machine
MLP	Multilayer Perceptron
SVM	Support Vector Machine
CNN	Convolutional Neural Network

LIST OF SYMBOLS

$\boldsymbol{X}$	The term-document matrix used in LSI
$\boldsymbol{X}_d$	A document in the term-document matrix of LSI, as a collumn vector
$\boldsymbol{T}_v$	A term of the corpus, as a row vector in the term-document matrix of LSI
$\hat{\boldsymbol{X}}$	An approximation to the original term-document matrix $\boldsymbol{X}$ of LSI, by maintaining only its largest singular values
$\boldsymbol{U}$	An orthonormal matrix with collumns containing the singular vectors of the term-document matrix $\boldsymbol{X}$, in LSI
$\boldsymbol{V}$	An orthonormal matrix with rows containing the singular vectors of the term-document matrix $\boldsymbol{X}$, in LSI
Σ	The diagonal matrix containing all the singular values of the term-document matrix $\boldsymbol{X}$, in LSI
$\hat{\Sigma}$	The diagonal matrix Σ with only the largest singular values of the term-document matrix $\boldsymbol{X}$, in LSI
$\hat{\boldsymbol{U}}$	The orthonormal matrix $\boldsymbol{U}$, with collumns containing only the singular vectors corresponding to the largest singular values of the term-document matrix $\boldsymbol{X}$, in LSI
$\hat{\boldsymbol{V}}$	The orthonormal matrix $\boldsymbol{V}$, with rows containing only the singular vectors corresponding to the largest singular values of the term-document matrix $\boldsymbol{X}$, in LSI
$\hat{\boldsymbol{X}}_d$	A document represented in the latent space found in LSI
$\hat{\boldsymbol{T}}_v$	A term of the corpus, represented in the latent space found in LSI
$\boldsymbol{q}$	A query submitted for categorization in, LSI
$\hat{\boldsymbol{q}}$	The representation of a query in the latent semantic space found in LSI
$\mathcal{D}$	A document collection
$\boldsymbol{N}$	The term document matrix of PLSI
$\mathcal{Z}$	The topic set in PLSI
$\boldsymbol{\beta}$	The matrix with word probabilities by topic, in LDA.
$\mathcal{M}(\boldsymbol{\beta}_k)$	The Multinomial distribution over words, for topic k in LDA
$\mathcal{M}(\boldsymbol{\theta}_d)$	The Multinomial distribution over topics, for document d, in LDA
$Dir(\boldsymbol{\alpha})$	The Dirichlet distribution from which the $\boldsymbol{\theta}_d$ vector is drawn, in LDA.
γ	The Dirichlet topic parameter of the variational distribution q, in LDA, for a single document

$\boldsymbol{\phi}$	The Multinomial parameter of the variational distribution q, in LDA, for a single document
γ^*	The optimal Dirichlet topic parameter of the variational distribution q, in LDA, for a single document
$\boldsymbol{\phi}^*$	The optimal Multinomial parameter of the variational distribution q, in LDA, for a single document
ψ	The first derivative of the $lof\Gamma$ function
η	The parameter of the Dirichlet distribution used to smooth the LDA model
λ	The Dirichlet parameter used in the variational distribution q, in LDA
μ_{ji}	The i-th component from the j-th Gaussian mean, in INBC
${\sigma^2}_{ji}$	The i-th component from the j-th Gaussian variance
τ_{nov}	A fraction of the maximum value of the likelihood function, in INBC
σ_{bl}	The baseline variance, in INBC
δ	A user-defined fraction of the overall variance of each attribute, in INBC
θ_j	The parameter vector of the j-th Multinomial distribution, in DV-INBC
δ	Fraction of the most frequent words inside a cluster, in DV-INBC
τ	Percentage of the maximum value assumed by the Jensen-Shannon distance, in DV-INBC

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The field of Data Mining has seen rapid advances in recent years, due to the availability of different kinds of data, which is particularly true for the case of text, where the web and social networks have enabled the rapid creation of large data repositories. The increasing amount of text data available from different applications has created a need for advances in algorithmic design which can learn interesting patterns from the data in a dynamic and scalable way (AGGARWAL; ZHAI, 2012). As our collective knowledge continues to be digitized and stored in the form of news, blogs, Web pages, scientific articles, books and many other types of media, computational tools to organize, search and understand such vast amounts of data are thus needed (BLEI, 2012).

In the case of text data, whether mining a text stream or a collection, the task of document classification and retrieval needs useful representations of the information contained in each document, and the fact that such information is originally available in an unstructured way motivates the research and design of algorithms capable of solving such problems (SRIVASTAVA; SALAKHUTDINOV; HINTON, 2013). Also, web applications (e.g. social networks) can result in a continuous stream of large volumes of text data, due to the simultaneous input of text from a wide variety of users. Such text data are more challenging to process, for they need to be processed in the context of a one-pass constraint, meaning that sometimes it may be difficult to store the data offline for processing and that the mining task should be performed as the data arrive (AGGARWAL; ZHAI, 2012).

A type of algorithms for discovering the main themes that pervade a collection of documents are the Topic Models. Those algorithms can organize the document collection according to the discovered themes. Also, they can be applied to massive collections of documents or even document streams (BLEI, 2012). The Topic Modeling area integrates soft clustering with dimension reduction. Documents are associated with a number of latent topics, which correspond to both document clusters and compact representations identified from a collection. Each document is assigned to each topic with different weights, which specify the degree of membership in each cluster. The original feature representation plays a key role in defining the topics and in identifying which topics are present in each document. The result is an understandable representation of documents that is useful for analyzing themes in documents (AGGARWAL; ZHAI, 2012).

Motivated by the need of algorithms capable of processing document streams and/or large collections, this work presents an approach to categorize documents that can be used on such scenarios: DV-INBC, which means Dynamic Vocabulary Incremental Naive Bayes Clustering. The method needs only information about the number of classes in the document collection, building the vocabulary dynamically as more data come in. The representation of each class in the model is a mixture of Multinomial distributions, following a similar approach of many Topic Model techniques. DV-INBC is an incremental, online and probabilistic algorithm that extends the INBC model presented in (ENGEL, 2009). Those characteristics of DV-INBC make it a suitable model for categorizing document streams and processing document collections as a continuous flow of data, as can be seen by the experiments presented in this thesis.

1.1 Main Contributions

The main contribution of this work is the development of a new algorithm for document classification, which is called DV-INBC, and its evaluation on popular datasets for this task. The results show that the model is promising although there are still improvements to be made. Also, the online algorithm presented in this thesis can be useful in scenarios where the data size is too big to be passed over many times as other techniques do, as well as in situations where there is a continuous document stream to be processed.

1.2 Structure of this text

Chapter 2 presents some Topic Model techniques for representing themes in document collections, and the classic Naive Bayes algorithm as an algorithm for document classification.

Chapter 3 presents the INBC algorithm and its main characteristics, which is the main basis of DV-INBC.

Chapter 4 presents the main topic of this thesis, the DV-INBC model. It shows the details of the model and its training and classification procedures.

Chapter 5 shows experiments made on popular document classification datasets and an analysis of the performance achieved by DV-INBC on those tests. The chapter is divided in two different tasks, on which DV-INBC has its performance compared to recent methods and to other popular algorithms for a simpler comparison.

Finally, Chapter 6 concludes this thesis with an analysis of the performance of DV-INBC and also presents some future works.

2 BACKGROUND

This chapter presents some Topic Model approaches to represent themes in document collections and techniques for classification of text data. Topic Model techniques are a set of algorithms to discover and annotate large collections of documents with thematic information. They use statistical methods to analyse the words of original texts to discover themes, how they are connected and how they change over time (BLEI, 2012). Also this chapter presents the classical Naive Bayes classification algorithm for text data, which was used in this thesis.

All of the techniques in this chapter use the *bag-of-words* (BOW) model to represent documents. In this representation, the order of the words in each document is ignored, and only their frequency[1] is kept. Also, it balances computational efficiency while retaining document content, resulting in a vector representation that can be analyzed by many Machine Learning techniques. However, since in BOW vectors each dimension corresponds to a different term in the vocabulary, those vectors can have a large number of dimensions. Since it is desirable to keep only the semantic space related to the topics in the document collection (which usually is much smaller than the vocabulary space), topic model techniques can be used to this end (AGGARWAL; ZHAI, 2012).

2.1 Latent Semantic Indexing (LSI)

The first methods for document retrieval and indexing used simple term-matching to select the most relevant documents to the query provided by the user. However, the main problem with this approach is that, often, the documents returned by the term-by-term comparison search engine would not be relevant for the user, due to synonymy[2]. Besides, documents that could be relevant would not be returned by the same reason. Therefore, even if a document was relevant to the user, it was not returned as a result by the engine if it did not contain the exact same words provided in the search query. The solution would be to find the latent semantics of the terms in the query and in each document. The Latent Semantic Indexing (LSI) method (DEERWESTER et al., 1990) was designed to overcome this problem, projecting the corpus onto a semantic space where it is easier to find semantically related terms, even when they were graphically different.

In LSI, a corpus is represented by a term-by-document matrix $\mathbf{X}$, where each document d is represented as collumn vector $\mathbf{X}_d$ and each row in $\mathbf{X}$, $\mathbf{T}_v$, denotes a term v of the corpus vocabulary. Each entry (v, d) in $\mathbf{X}$ is, therefore, the frequency of term v in document d.

To find the latent semantics of the corpus, LSI performs a Singular Value Decomposition (SVD) over $\mathbf{X}$, maintaining only its K largest singular values. The decomposition of $\mathbf{X}$ is then

[1] Frequency in the statistical sense, which means the number of occurrences of a specific event.

[2] The property of different words having the same meaning.

defined as

$$\hat{\boldsymbol{X}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^T = [\boldsymbol{U}_1 \cdots \boldsymbol{U}_K] \cdot \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{V_1}^T \\ \vdots \\ \boldsymbol{V_K}^T \end{bmatrix} \tag{2.1}$$

where $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ are orthonormal matrices, $\hat{\boldsymbol{\Sigma}}$ is diagonal and $\hat{\boldsymbol{X}}$ is an approximation to the original matrix $\boldsymbol{X}$.

The decomposition in (2.1) produces the rank-K matrix $\hat{\boldsymbol{X}}$ with the best least-squares-fit to $\boldsymbol{X}$. Also, $\hat{\boldsymbol{X}}$ can be viewed as a smoothed version of $\boldsymbol{X}$, which is achieved by discovering the latent semantic space formed by the documents (AGGARWAL; ZHAI, 2012). In this K-dimensional space, each dimension is interpreted as a latent topic. The choice of K is important because it adjusts the amount of admissible error in the model: a large value can assimilate noise as a latent semantics and a small value can miss important features in the data (DEERWESTER et al., 1990).

A document d in term space $\boldsymbol{X}_d$ can be represented using the latent semantic space as

$$\boldsymbol{X}_d = \hat{\boldsymbol{U}} \cdot \hat{\boldsymbol{\Sigma}} \cdot \hat{\boldsymbol{X}}_d \tag{2.2}$$

and, similarly, each term v can be represented by the K-dimensional vector $\hat{\boldsymbol{T}}_v$ given by

$$\boldsymbol{T}_v = \hat{\boldsymbol{V}} \cdot \hat{\boldsymbol{\Sigma}} \cdot \hat{\boldsymbol{T}}_v. \tag{2.3}$$

The similarity between documents can be measured through their representations in the new semantic space using, for instance, the inner product between their projections. This can also be used to cluster or categorize documents. Term similarity can also be measured in an analogous manner.

When performing a document retrieval task, it is necessary to represent a query $\boldsymbol{q}$ using the latent semantic space to compare it with other documents and select the most similar ones. The representation is derived from (2.2), by considering $\boldsymbol{X}_d$ as the query $\boldsymbol{q}$:

$$\hat{\boldsymbol{q}} = \hat{\boldsymbol{\Sigma}}^{-1} \cdot \hat{\boldsymbol{U}}^T \cdot \boldsymbol{q}. \tag{2.4}$$

To handle changes in the corpus, some modifications to the original algorithm need to be applied. A simple way is to represent the new documents using the SVD decomposition from the previous set of documents. This is a very efficient method, since it is not necessary to recompute the SVD. However, as more documents are added, there is no guarantee that the semantics space still represents the corpus adequately, thus an update to the decomposition is necessary either periodically or at every new document (AGGARWAL; ZHAI, 2012).

2.2 Probabilistic Latent Semantic Indexing (PLSI)

Following similar concepts to LSI, Probabilistic Latent Semantic Analysis (PLSI) (HOFMANN,) uses a semantic space to represent the document collection. The main difference is that its approach is probabilistic, defining a generative model for the document collection.

PLSI organizes a collection composed by a set $\mathcal{D}$ of N documents, described by a vocabulary $\mathcal{W}$ of M possible words as a $N \times M$ term-document matrix with elements $\boldsymbol{N}_{ij} = n(d_i, w_j)$, where $n(d_i, w_j) \in \mathbb{N}$ indicates how many times term w_j occurred in document d_i. Also, a Multinomial probability distribution over a latent variable $z \in \mathcal{Z} = z_1, \cdots, z_K$ is associated with each document of the collection, defining the probability of each topic for each document. The joint probability model for words and documents is defined in (HOFMANN,) as

$$P(d, w) = P(d) \cdot P(w|d), \tag{2.5}$$

where

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z) \cdot P(z|d). \tag{2.6}$$

The generative process defined in PLSI for a *token* (a word) w in document d can be described as:

- sample a document d from a Multinomial distribution $P(d)$;
- sample a topic $k \in 1, \cdots, K$ based on the topic distribution $P(z = k|d)$
- sample a word v for token w based on $P(w = v|z = k)$.

This process assumes that the probability distribution of terms conditioned on documents $P(w|d)$ is a convex combination of the topic-specific term distributions $P(w = v|z = k)$ (AGGARWAL; ZHAI, 2012).

(HOFMANN,) also defines the joint probability $P(d, w)$, in an equivalent manner, as

$$P(d, w) = \sum_{z \in \mathcal{Z}} P(z) \cdot P(d|z) \cdot P(w|z), \tag{2.7}$$

which models documents and terms in a symmetric manner, conditioned on the topic z. This formulation associates to each observation (d, w) the unobservable topic variable z and is based on a statistical model called the *aspect model*, which is the main basis of PLSI (HOFMANN,). Also, this formulation helps to see the method as a probabilistic analogue to LSI, where matrices $\hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{U}}$ function as the distributions $P(d|z)$ and $P(w|z)$, acting as the projections of documents and terms into the latent semantic space. Also, the distribution $P(z)$ is similar to the diagonal matrix $\hat{\boldsymbol{\Sigma}}$ in LSI, that acts as the weight of each topic in the collection.

During training, to estimate the model parameters $P(d)$, $P(z|d)$ and $P(w|z)$, the EM algorithm (DEMPSTER; LAIRD; RUBIN, 1977) is used, maximizing the log-likelihood of the training data, defined as

$$\mathcal{L} = \sum_{d=1}^{M} \sum_{w \in \mathcal{W}} n(d, w) \cdot \log \sum_{z \in \mathcal{Z}} P(w|z) \cdot P(z|d). \tag{2.8}$$

The expectation step of EM computes the posterior of the latent variable z based on the current parameters:

$$p'(z = k|d, w = v) = \frac{p(d) \cdot p(z = k|d) \cdot p(w = v|z = k)}{\sum_{k \in \mathcal{Z}} p(d) \cdot p(z = k|d) \cdot p(w = v|z = k)}, \tag{2.9}$$

which must iterate over all possible words $v \in \mathcal{W}$. The maximization step updates all parameters after knowing the latent variables by using the previous stage posteriors:

$$p'(w = v|z) \propto \sum_{d \in \mathcal{D}} n(d, v) \cdot p'(z = k|d, w = v), \tag{2.10}$$

$$p'(z = k|d) \propto \sum_{v \in \mathcal{D}} n(d, v) \cdot p'(z = k|d, w = v), \tag{2.11}$$

and

$$p'(d) \propto \sum_{v \in \mathcal{W}} n(d, v). \tag{2.12}$$

In all equations, $p'(\cdot)$ indicates the new value for that parameter.

However, according to (BLEI; NG; JORDAN, 2003), PLSI has a great risk of overfitting, because the number of parameters in the model grows linearly with the size of the corpus: for a document collection of size N, a set of K topics and a vocabulary of M words, there is $K \cdot N + K \cdot M$ parameters to adjust. Therefore, some techniques have to be used in order to speed up training and reduce the risk of overfitting. (HOFMANN,) suggests to use an alternative heuristic approach for training, called a “tempered” version of EM. Nevertheless, the model still can get overfitted to the training data.

2.3 Latent Dirichlet Alocation (LDA)

Latent Dirichlet Alocation (LDA) (BLEI; NG; JORDAN, 2003) is a probabilistic generative model for documents, greatly reducing the number of parameters to be adjusted (when compared to PLSI) and providing a clearly-defined form of assigning probabilities for arbitrary documents outside the training set (AGGARWAL; ZHAI, 2012).

LDA improves over the two major problems of PLSI, which are:

- a linear growth in the number of parameters, with the size of the corpus (leading to overfitting) and;

- the lack of clarity on how to assign probabilities to documents outside the training set, since the topic probability distributions $P(z|d)$ are learned only for the training documents.

In LDA, documents are represented as mixtures over a set of k latent topics. Each topic is defined as a Multinomial distribution $\mathcal{M}(\boldsymbol{\beta_k})$ parameterized by a $\boldsymbol{\beta_k}$ vector, over a vocabulary of size V. In (BLEI; NG; JORDAN, 2003), this distribution is represented as a $k \times V$ matrix $\boldsymbol{\beta}$ (where V is the vocabulary size) and $\beta_{ij} = p(w^j|z^i)$.

Each document has its own probability distribution over topics, defined as another Multinomial distribution $\mathcal{M}(\boldsymbol{\theta_d})$ parameterized by a $\boldsymbol{\theta_d}$ vector, drawn from a Dirichlet distribution parameterized by an $\boldsymbol{\alpha}$ vector. The $\boldsymbol{\alpha}$ vector and the $\boldsymbol{\beta}$ matrix are the only corpus-level parameters, the $\boldsymbol{\theta}_d$ vectors are sampled once for each document and all others are sampled repeatedly for each word therein.

Considering the d-th document $\mathbf{w}$ with N words, the generative process of LDA is defined as:

- sample a $\boldsymbol{\theta}_d$ vector from a Dirichlet distribution $Dir(\boldsymbol{\alpha})$
- for each word w_{dn} in the document, sample a topic z_{dn} from $\mathcal{M}(\boldsymbol{\theta_d})$ and then sample word w_{dn} from the multinomial probability distribution $\mathcal{M}(\boldsymbol{\beta_k})$ conditioned on the topic z_{dn}.

For the d-th document, the joint probability of a topic mixture $\boldsymbol{\theta}_d$, a set of N topics $\mathbf{z}_d$, and a set of N words $\mathbf{w}_d$ is given by:

$$p(\boldsymbol{\theta}_d, \mathbf{z}_d, \mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \cdot \prod_{n=1}^{N} p(z_{dn}|\boldsymbol{\theta}_d) \cdot p(w_{dn}|z_{dn}, \boldsymbol{\beta}), \tag{2.13}$$

where $p(z_{dn}|\boldsymbol{\theta}_d)$ is simply θ_{id} for the unique i such that the i-th position of the z_{dn} vector is 1. Summing over all possible values of the z variable and integrating over $\boldsymbol{\theta}_d$ the marginal distribution of a single document is:

$$p(\mathbf{w}_d|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \cdot \left(\prod_{n=1}^{N} \sum_{z_{dn}} p(z_{dn}|\boldsymbol{\theta}_d) \cdot p(w_n|z_{dn}, \boldsymbol{\beta}) \right) \mathrm{d}\boldsymbol{\theta}_d. \tag{2.14}$$

Finally, the probability of a corpus $\mathcal{D}$ with M documents is obtained by multiplying over all documents:

$$p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{M} \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \cdot \left(\prod_{n=1}^{N} \sum_{z_{dn}} p(z_{dn}|\boldsymbol{\theta}_d) \cdot p(w_n|z_{dn}, \boldsymbol{\beta}) \right) \mathrm{d}\boldsymbol{\theta}. \tag{2.15}$$

The main difference between LDA and a simple Dirichlet-multinomial clustering model regards the number of topics that a document can be associated to. In LDA, after sampling a

Dirichlet distribution for the corpus, a Multinomial distribution is sampled for each word in each document, which allows a single document to be associated with different topics. Conceptually, this means that a single document is composed of words that were gererated by different topics, which is very similar to what really happens on texts. On the other hand, a simple Dirichlet-multinomial model samples a single Multinomial distribution for each document, which assumes that each document was generated by repeatedly sampling words from a distribution conditioned on the topic variable, i.e. from only one topic at a time (BLEI; NG; JORDAN, 2003).

However, according to (AGGARWAL; ZHAI, 2012), LDA has the disadvantage of learning broad topics. In a situation where a concept has a number of aspects to it and each of the aspects co-occurs frequently with the main concept, LDA will favor a topic that includes the concept and all of its aspects. It will further favor adding other concepts to the same topic if they share the same aspects. As this process continues, the topics become more diffuse. When sharper topics are desired, a hierarchical topic model may be more appropriate.

To compute the posterior probability of the topic structure of the model given an observed document, the following equation must be solved:

$$p(\boldsymbol{\theta_d}, \mathbf{z}_d | \mathbf{w}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}_d) = \frac{p(\boldsymbol{\theta_d}, \mathbf{z}_d, \mathbf{w}_d | \boldsymbol{\alpha}, \boldsymbol{\beta}_d)}{p(\mathbf{w}_d | \boldsymbol{\alpha}, \boldsymbol{\beta}_d)}. \tag{2.16}$$

However, the denominator $p(\mathbf{w}_d | \boldsymbol{\alpha}, \boldsymbol{\beta}_d)$ is intractable to compute. This is due to two facts:

- the exponentially large number of possible topic structures that could generate the current document and;
- the coupling between the $\boldsymbol{\theta}_d$ and $\boldsymbol{\beta}$ variables, when marginalizing over the latent topics.

Those two problems can be seen when (2.14) is rewritten in terms of all the model parameters:

$$p(\mathbf{w}_d | \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(\sum_i \boldsymbol{\alpha})}{\prod_i \Gamma(\boldsymbol{\alpha}_i)} \int \left(\prod_{i=1}^{k} \theta_{di}^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_{di} \beta_{ij})^{{w_n}^j} \right) \mathrm{d}\theta_d, \tag{2.17}$$

where $\Gamma(\cdot)$ is the Gamma function, used in the Dirichlet distribution from which the $\boldsymbol{\theta}_d$ vectors are sampled from:

$$p(\boldsymbol{\theta}_d | \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} {\boldsymbol{\theta}_1}^{\alpha_i - 1} \ldots {\boldsymbol{\theta}_k}^{\alpha_k - 1}. \tag{2.18}$$

To train an LDA model, its parameters must be optimized to maximize the probability of generating the training data. However, direct optimization of those parameters is intractable, due to the two problems mentioned above. To deal with this situation, variational approaches as well as sampling techniques[3] are used (AGGARWAL; ZHAI, 2012). A wide variety of approximate

[3] Variational approaches and sampling techniques are usually applied to estimate the value of integrals in

inference algorithms can be considered for LDA, including Laplace approximation, variational approximation and Markov chain Monte Carlo (BLEI; NG; JORDAN, 2003).

In (BLEI; NG; JORDAN, 2003) it is presented a variational inference method for LDA, which uses the EM algorithm. In this method, the relations between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ that are problematic are eliminated, which translates into a simpler graphical model, shown in Figure 2.1. The variational distribution used to approximate the true posterior distribution of LDA has the form, for a single document:

$$q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \boldsymbol{\phi}) = q(\theta|\gamma) \cdot \prod_{n=1}^{N} q(z_n|\phi_n), \tag{2.19}$$

where the Dirichlet paramter γ and the Multinomial parameters $(\phi_1, \cdots, \phi_N)$ are the free variational parameters.

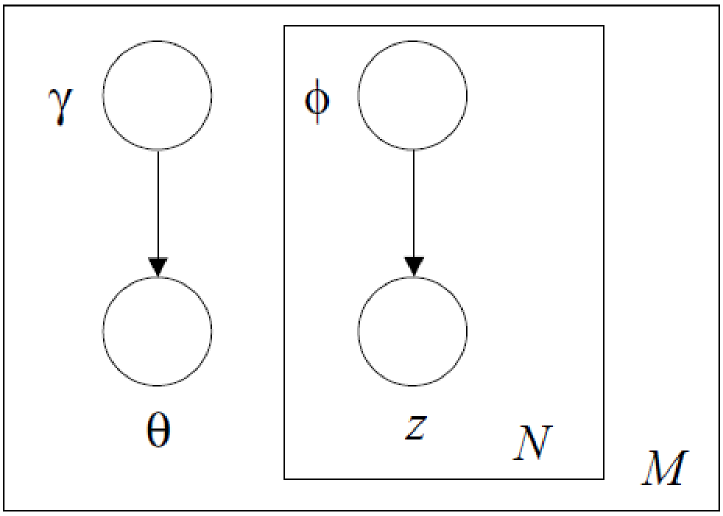


Figure 2.1: A graphical model representation of the variational distribution used to approximate the posterior in the LDA model. Adapted from (BLEI; NG; JORDAN, 2003).

The optimal parameters γ^* and ϕ^* for a single document are obtained by an iterative fixed-point method, using the following pair of equations:

$$\phi_{ni} \propto \beta_{iw_n} \cdot exp\left\{E_q[\log(\theta_i)|\gamma]\right\} \tag{2.20}$$

and

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}, \tag{2.21}$$

where $E_q[\log(\theta_i)|\gamma] = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{K} \gamma_j)$ and $\Psi(\cdot)$ is the first derivative of the $\log \Gamma$ function, which, as stated in (BLEI; NG; JORDAN, 2003), can be computed via Taylor approximations.

Bayesian inference or approximate the value of the marginal data likelihood, which can be analytically intractable, depending on the model. A variational method yields an analytical approximation but usually requires more work to derive the set of equations that iteratively update the parameters, while a sampling technique yields an approximate numerical solution, albeit being easier to find the sampling equations.

Each ϕ_{ni} is defined as proportional to $\beta_{iw_n} \cdot exp\left\{E_q[\log(\theta_i)|\gamma]\right\}$ because they should be normalized so $\sum_i \phi_{ni} = 1$ is true.

Equation (2.21) updates the Dirichlet topic parameter γ and is a posterior Dirichlet, given the expected observations taken under the variational distribution $E[z_n|\phi_n]$. The update performed in (2.20) essentially uses Bayes's theorem, $p(z_n|w_n) \propto p(w_n|z_n)p(z_n)$, where $p(z_n)$ is approximated by the exponential of the expected value of its logarithm under the variational distribution.

The E-step of the EM algorithm consists on finding the optimal parameters γ^* and ϕ^* for all documents. On the M-step, the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are found, by maximizing the lower bound on the log-likelihood of the data, using the variational distribution (2.19). The update process for the $\boldsymbol{\beta}$ parameter is

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N} {\phi^*}_{dni} {w_{dn}}^j \tag{2.22}$$

which is much simpler than the one used to update $\boldsymbol{\alpha}$, that uses a Newton-Rhapson algorithm to determine the optimal $\boldsymbol{\alpha}$. The complete training algorithm for training an LDA model can be found in (BLEI; NG; JORDAN, 2003).

Since the vocabulary size of the corpus is much larger than the set of different words found in a document, when optimizing the model parameters, the $\boldsymbol{\beta}$ matrix can have some words with a probability equal to zero. This problem can be solved by placing a Dirichlet prior over $\boldsymbol{\beta}$ and assuming that it is a random matrix, with each row independently sampled from an exchangeable Dirichlet distribution with parameter η. This leaves the model with two hyperparameters: $\boldsymbol{\alpha}$ and η. The resulting smoothed LDA model is shown in Figure 2.2, and the complete variational training algorithm can be found in (BLEI; NG; JORDAN, 2003).

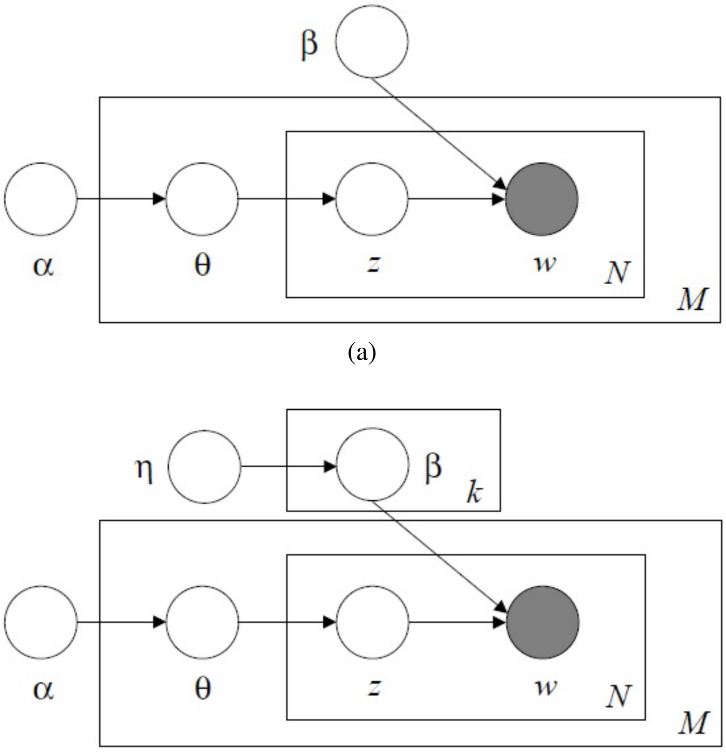
As a consequence of such smoothing, the variational equations also need to be changed, with the addition of another parameter, λ:

$$q(\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\theta}|\lambda, \phi, \gamma) = \prod_{i=1}^{K} Dir(\beta_i|\gamma_i) \prod_{d=1}^{M} q_d(\theta_d, \boldsymbol{z}_d|\phi_d, \gamma_d), \tag{2.23}$$

where $q_d(\boldsymbol{\theta}, \boldsymbol{z}|\phi, \gamma)$ is the variational distribution defined in (2.19). The update equations for parameters ϕ and γ stay the same, and a new one for λ is derived:

$$\lambda_{ij} = \eta + \sum_{d=1}^{M} \sum_{n=1}^{N_d} {\phi^*}_{dni} {w^j}_{dn}. \tag{2.24}$$

There are some improvements over the traditional LDA model regarding online capabilities and inference algorithms for training. An online variational training method for LDA is presented in (HOFFMAN; BACH; BLEI,), allowing the model to fit a fixed number of topics to document streams. In (WANG; PAISLEY; BLEI,) an online approach for LDA is presented, where Hierarchical Dirichlet Processes are used to decide the optimal number of topics to fit



(a)

(b)

Figure 2.2: The two versions of the LDA model: (a) shows the original model, with no smoothing on the multinomial $\boldsymbol{\beta}$ matrix and (b) depicts the smoothed model, with a Dirichlet distribution $Dir(\eta)$ as a prior on the multinomial. Adapted from (BLEI; NG; JORDAN, 2003).

during runtime.

2.4 Naive Bayes

The Naive Bayes classifier is a highly practical method. It is applied to learning tasks where each instance $\boldsymbol{x}$ is described by a conjunction of attribute values and where the instance label can take on any value from a finite set of labels V. It is a supervised learning method (MITCHELL, 1997).

The method assigns the most probable label v_{MAP}, given the attribute values $a_1, a_2, \cdots, a_n$ that describe $\boldsymbol{x}$.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2, \cdots, a_n). \tag{2.25}$$

Following the Bayes therorem, (2.25) can be rewritten as

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, \cdots, a_n | v_j) \cdot P(v_j)}{P(a_1, a_2, \cdots, a_n | v_j)}, \tag{2.26}$$

which is equivalent to

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, \cdots, a_n | v_j) \cdot P(v_j). \tag{2.27}$$

The two terms in (2.27) can easily be estimated from the training data. To learn the value of $P(v_j)$, it is possible to simply count the frequency of each target value v_j in the training set. Besides, since the Naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given the instance label, the probability of observing the conjunction $a_1, a_2, \cdots, a_n$ can be modelled as the product of the probabilities for the individual attributes:

$$P(a_1, a_2, \cdots, a_n | v_j) = \prod_i P(a_i | v_j). \tag{2.28}$$

Finally, by substituting (2.28) into (2.26) the approach used by the Naive Bayes classifier can be defined as

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j), \tag{2.29}$$

where v_{NB} denotes the output value of the classifier.

If using this classifier for text, each data instance $\boldsymbol{x}$ would be a document, where each attribute a_i would indicate the frequency of a word inside that document. Finally, the label v_j would be the topic associated to each document of the collection and could be estimated in the same manner.

2.5 Conclusions

This chapter presented four approaches that can be used in document classification and topic modeling: Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (PLSI), Latent Dirichlet Alocation (LDA) and the Naive Bayes classifier. The first three methods are used to identify the latent topics of a document collection and are all unsupervised methods, finding a latent space to represent the original data. This means that by setting a total number of K topics to find, the algorithms yield a subspace of that same amount of dimensions, allowing to use those new representations to train another classifier to learn how to categorize documents of that same collection.

The Naive Bayes classifier is a traditional method used in categorization problems and has a very simple model. It uses the assumption that the set of attributes of a data point $\boldsymbol{x}$ is conditionally independent, given the label of the point. It allows a fast computation of the model parameters by iterating only once over the training data and adjusting its two main parameters, that indicate the prior probability of each class and the probability of each word inside each class.

3 THE INCREMENTAL NAIVE BAYES CLUSTERING ALGORITHM

The Incremental Naive Bayes Clustering (INBC) algorithm was initially presented in (ENGEL, 2009) and is based on a probabilistic framework, using a mixture of Gaussian distributions to describe a data stream. INBC follows an unsupervised incremental learning paradigm, where each data point is just instantaneously available to the learning system. In this situation, it is necessary to take into account the instantaneous data to update the model of the environment, since after its use the point is unavailable. Also, since INBC adopts an incremental mixture distribution model, it dynamically controls the number of mixture components that represent the so far presented data.

3.1 The probabilistic framework of INBC

Considering an input data vector $\mathbf{x} \in \mathbb{R}^d$, in INBC it is assumed that its probability density can be modeled by a linear combination of *M* component probability densities $p(\mathbf{x}|j)$

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x}|j) \cdot p(j), \tag{3.1}$$

where each coefficient $p(j)$ is called a mixing parameter. Each of those parameters is related to the prior probability of its respective component generating **x**.

The "naive" assumption followed by INBC means that each attribute of the data points are all conditionally independent, given component j. This means that the probability of the conjunction of attributes in each component distribution is the product of the probabilities of each one, individually. Therefore, the probability of the j-th mixture component generating the data vector $\mathbf{x} = (x_1, \cdots, x_i, \cdots, x_d)$ is computed as the product $p(\mathbf{x}|j) = \prod_i^d p(x_i|j)$.

Each component density $p(x_i|j)$ is modeled by a unidimensional normal Gaussian distribution function, of the form

$$p(x_i|j) = \frac{1}{\sqrt{2\pi{\sigma_{ji}}^2}} \exp\left\{-\frac{(x_i - \mu_{ji})^2}{\sigma_{ji}^2}\right\}, \tag{3.2}$$

where μ_{ji} is the i-th component from the j-th Gaussian mean and σ_{ji}^2 is the i-th component from the j-th Gaussian variance. The posterior probability of each component is given by

$$p(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)p(j)}{\sum_{j=1}^{M} p(\mathbf{x}|j)p(j)}. \tag{3.3}$$

3.2 Component management

A new component of the mixture is created if the input vector **x** matches the *novelty criterion* defined as

$$p(x_i|j) < \frac{\tau_{nov}}{\sqrt{2\pi{\sigma_{ji}}^2}} \qquad \forall i, \forall j, \tag{3.4}$$

where τ_{nov} is a fraction of the maximum value of the likelihood function. The criterion follows a minimum likelihood approach, that all components from **x** must fulfill. This criterion defines that the **x** vector must minimally fit at least one mixture component. If the likelihood $p(x_i|j)$ of at least one attribute of the vector **x** is greater than the threshold defined by the *novelty criterion*, the model parameters are updated, and no new component is created. The novelty criterion is therefore evaluated as false.

When the data point **x** is not considered a "novelty" by INBC, the model parameters are updated by an online version of the EM algorithm. The *estimation* (E) step is done by computing the posterior probability of each component membership for the data point, which is obtained by (3.3). The result can be used to estimate new values for the model parameters, in the *maximization* (M) step. Each new value for the mean ${\mu_{ji}}^{new}$, variance ${\sigma^2}_{ji}^{new}$ and prior probability of each component $p(j)^{new}$ are obtained by the following equations

$${\mu_{ji}}^{new} = \frac{\sum_{n=1}^{N} p^{old}(j|\boldsymbol{x}^n) \cdot {x_i}^n}{\sum_{n=1}^{N} p^{old}(j|\boldsymbol{x}^n)}, \tag{3.5}$$

$${\sigma^2}_{ji}^{new} = \frac{\sum_{n=1}^{N} p^{old}(j|\boldsymbol{x}^n) \cdot ({x_i}^n - {\mu_{ji}}^{new})^2}{\sum_{n=1}^{N} p^{old}(j|\boldsymbol{x}^n)} \tag{3.6}$$

and

$$p(j)^{new} = \frac{1}{N} \cdot \sum_{n=1}^{N} p^{old}(j|\boldsymbol{x}^n), \tag{3.7}$$

where *N* is the number of data points presented so far.

The model update process can be written using a set of accumulator variables, in order to simplify the procedure. At every new data point, INBC updates the sp_j variable, adding to its value the posterior probability of the j-th component

$$\sum_{n=1}^{t} p(j|\mathbf{x}^n) = p(j|\boldsymbol{x}^t) + \sum_{n=1}^{t-1} p(j|\boldsymbol{x}^n) \tag{3.8}$$

which can be divided in the *new* and *old* values of sp_j:

$${sp_j}^{new} = p(j|\boldsymbol{x}^t) + {sp_j}^{old} \tag{3.9}$$

also, ${sp_j}^{new}$ can be used to compute $p(j)^{new}$

$$p(j)^{new} = \frac{1}{t} \cdot \sum_{n=1}^{t} p(j|\boldsymbol{x}^n) = \frac{1}{t} \cdot {sp_j}^{new}, \tag{3.10}$$

where the current number of presented data points t can be written as $t = \sum_{j=1}^{M} sp_j$ and therefore the final form of equation (3.10) is:

$$p(j)^{new} = \frac{{sp_j}^{new}}{\sum_{j=1}^{M} {sp_j}^{new}} \tag{3.11}$$

Another accumulator, used in the update of the mean ${\mu_j}^{new}$ of each component, is called ${spx_{ji}}^{new}$, and is computed for each attribute of the data vectors. Its values are the sum of products of posterior probabilities for component j by the respective values of attribute i for all points presented so far. The current data point $\boldsymbol{x}^t$ contributes to this sum with the product of its posterior probability to the component j by the value of its attribute i, as written in the equation below:

$${spx_{ji}}^{new} = p(j|\boldsymbol{x}^t) \cdot {x_i}^t + {spx_{ji}}^{old}. \tag{3.12}$$

Then, using sp_j and spx_{ji} each attribute of the mean of each component of the mixture is updated as:

$${\mu_{ji}}^{new} = \frac{p(j|\boldsymbol{x}^t){x_i}^t + \sum_{n=1}^{t-1} p(j|\boldsymbol{x}^n){x_i}^n}{p(j|\boldsymbol{x}^t) + \sum_{n=1}^{t-1} p(j|x_n)} = \frac{{spx_{ji}}^{new}}{{sp_j}^{new}}. \tag{3.13}$$

Finally, to update the attributes of the variance of each component, equation (3.6) is used. However, it can be rewritten using an accumulator as well:

$${\sigma^2}_{ji}^{\ new} = \frac{p(j|\boldsymbol{x}^t)({x_i}^t - \mu_{ji}^{new})^2 + \sum_{n=1}^{t-1} p(j|\boldsymbol{x}^n)({x_i}^n - \mu_{ji})^2}{p(j|\boldsymbol{x}^t) + \sum_{n=1}^{t-1} p(j|\boldsymbol{x}^n)} = \frac{sps_{ji}^{new}}{sp_j^{new}} \tag{3.14}$$

If the novelty criterion considers the data point **x** a “novelty”, a new component M is created, as a Gaussian distribution centered at **x** and with the *baseline variance* ${\sigma_b}^2$. This variance is computed as a user-defined fraction of the overall variance of each attribute of the input, before training

$${\sigma_{bi}}^2 = \delta_i \left[max(x_i) - min(x_i)\right]^2. \tag{3.15}$$

The initialization of the parameters of the new component is done following the equations below.

$$\mu_{M,i} = x_i \quad \forall i \tag{3.16}$$

$$\sigma^2_{M,i} = \sigma^2_{bi} \quad \forall i \tag{3.17}$$

$$sp_M = 1, \tag{3.18}$$

$$spx_M = 0, \tag{3.19}$$

$$sps_M = 0 \tag{3.20}$$

and

$$p(M) = \frac{1}{\sum_{j=1}^{M} sp_j}. \tag{3.21}$$

Besides creating new components as needed, INBC has a mechanism to identify and remove spurious ones. For this end, every component has an age attribute which is used together with its sp accumulator. A spurious component is defined as one that has an age attribute greater than an age_{min} parameter and an sp value less than an sp_{min} threshold. If, for any component in the model those two conditions are true, it is removed. Every component starts with $age_j = 1$ and at every new data vector this value is increased by 1.

The INBC algorithm for every new data point **x** is presented in Algorithm 1.

3.3 Conclusion

This chapter presented the INBC algorithm, which is the basis of this thesis. It is an algorithm capable of grouping streaming data into clusters, defined by a Gaussian distribution. The resulting model is therefore a Gaussian mixture, which can be used to perform a probabilistic classification of data into its clusters. The main characteristics of INBC, which are its capability of changing its inner structure (adding and removing clusters) in an online setting make it an attractive model to the task that this thesis approaches, the categorization of document streams.

Algorithm 1 INBC algorithm

```
input: a new data vector x
{Computes the likelihood of x to all components}
for $j = 1$ to $M$:
  $p(\mathbf{x}|j) = \prod_{i=1}^{d} p(x_i|j)$
end for
{Checks the novelty criterion}
if $p(x_i|j) < \frac{\tau_{nov}}{\sqrt{2\pi{\sigma_{ji}}^2}} \quad \forall i, \forall j$:
  {Creates a new component}
  $\mu_{M,i} = x_i \quad \forall i$
  $\sigma^2_{M,i} = \sigma^2_{bi} \quad \forall i$
  $sp_M = 1$
  $sps_M = 0$
  $spx_M = 0$
  $p(M) = \frac{1}{\sum_{j=1}^{M} sp_j}$
  $age_j = 1$
end if
for $j = 1$ to $M$:
  $p(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)p(j)}{\sum_{j=1}^{M} p(\mathbf{x}|j)p(j)}$
end for
{Updates components parameters}
$sp_j^{new} = {sp_j}^{old} + p(j|\mathbf{x})$
${spx_{ji}}^{new} = {spx_{ji}}^{old} + p(j|\mathbf{x}) \cdot x_i$
${sps_{ji}}^{new} = {sps_{ji}}^{old} + p(j|\mathbf{x}) \cdot (x_i - \mu_{ji})^2$
$\mu_{ji}^{new} = \frac{{spx_{ji}}^{new}}{{sp_j}^{new}}$
$\sigma_{ji}^{2\ new} = \frac{{sps_{ji}}^{new}}{{sp_j}^{new}}$
$p(j)^{new} = \frac{sp_j}{N}$
$age_j = age_j + 1$
{Removes any spurious component}
for $j = 1$ to $M$:
  if $age_j > age_{min}$ and $sp_j < sp_{min}$ :
    {Removes component $j$}
  end if
end for
```

4 DYNAMIC VOCABULARY INBC: DV-INBC

The Dynamic Vocabulary INBC is an extension to the INBC algorithm, allowing its vocabulary to grow as new documents are processed. It is based on the probabilistic framework of INBC, with the major differences being the type of probability distribution used to model the data and the similarity criterion between components and data vectors. Also, DV-INBC is an incremental model that adapts its structure to better represent the data flow as new points are presented to it.

To represent each topic in the document collection, a different model for each one is created. Then, documents of that topic are presented to the respective model and clusters are generated and adjusted to learn sub-topics. At any moment, for each topic, there will be a configuration of clusters that best represent the document collection at that point. The whole training process will be presented in details in this chapter.

4.1 The model

Each input document $\boldsymbol{d}_n$ is represented by a *BOW* (*bag-of-words*) vector. In this representation, each dimension of the vector is related to a different word of the vocabulary used to represent that document. Usually, a single vocabulary is used to represent all documents, which makes those vectors very sparse i.e. have many zeroes. DV-INBC assumes a dynamic vocabulary, therefore each new document is represented by a vector of the form

$$\boldsymbol{d}_n = \left(f_{d_n w_1}, \cdots, f_{d_n w_i}, \cdots, f_{d_n w_{|V_n|}} \right), \tag{4.1}$$

where $f_{d_n w_i}$ is the frequency for word w_i inside document $\boldsymbol{d}_n$, and $|V_n|$ is the size of V_n, the vocabulary found in document d_n (the set of different words found inside d_n). The different vocabulary sizes found in each document are a consequence of the fact that each text can approach different themes, and so it is not expected to find the exact same vocabulary in all documents. Also, the same theme may be approached by different words.

A DV-INBC model is composed by a set of clusters, defined by centroids of the kind

$$\boldsymbol{c}_j = \left(f_{j w_1}, \cdots, f_{j w_i}, \cdots, f_{j w_{|V_j|}} \right), \tag{4.2}$$

where $f_{j w_i}$ is the frequency of word w_i inside cluster j and $|V_j|$ is the amount of different words found in that cluster i.e. its vocabulary. As documents are presented, the centroid of each cluster is adapted by adding new words or updating the frequency of the old ones.

The word frequencies in each cluster are used to define Multinomial distributions, parameterized by a vector of the kind

$$\boldsymbol{\theta}_j = (\theta_{j1}, \cdots, \theta_{ji}, \cdots, \theta_{j|V_j|}), \tag{4.3}$$

where θ_{ji} is the probability of sampling word w_i from cluster j. To compute each θ_{ji}, the following equation is used

$$\theta_{ji} = \frac{f_{jw_i}}{\mid \boldsymbol{c}_j \mid}, \tag{4.4}$$

where f_{jw_i} is the frequency of word w_i inside cluster j and $|\boldsymbol{c}_j| = \sum_{i=1}^{|V_j|} f_{jw_i}$, which is the *size* of cluster j. The summation $\sum_{i=1}^{|V_j|} \theta_{ji} = 1$, for all clusters.

As in INBC, the probability of observing a vector $\boldsymbol{d}_n$ is a linear combination of component densities, weighted by the priors of each one. In DV-INBC, the likelihood for a particular document $\boldsymbol{d}_n$ given a cluster j, $p(\boldsymbol{d}_n|j)$, follows a Naive Bayes approach. However, since the model does not assume a fixed and unique vocabulary for clusters and documents, some modifications to compute it were made. The process is explained in details in section 4.3. Also, in an analogous way, as INBC learned a Gaussian mixture from the data flow, a DV-INBC model learns a Multinomial mixture, where each component is a cluster.

4.2 DV-INBC as a Generative Model

During training, DV-INBC learns a Multinomial mixture by creating and adapting clusters as probability distribution over words, to represent document topics. Therefore, it can also be presented as a generative model of documents, creating them from what it has learned. To this end, the steps below should be followed:

- choose a model i.e. a topic;
- for each word, choose a cluster inside the model and generate a word from it.

The generative process of DV-INBC assumes that each document has words sampled from only one topic: just one model is sampled and then it generates all the words of the document. This approach has a lower representative power than PLSI and LDA, since a real document can have words from many topics (although it is expected to find more words from the main subject of that document). DV-INBC follows this assumption because it was based on the INBC model, which has a very similar generative model as can be seen in (3.1). Therefore, since DV-INBC extends INBC, this limitation is an expected consequence.

When compared to DV-INBC, it can be seen that LDA has a more adequate way of representing topics by using a Bayesian approach to learn them. When compared to PLSI, DV-INBC uses a different representation of topics, since it does not associate a Multinomial distribution to each document (which was a major cause of overfitting in that model). However, by sampling a single topic to generate each document, DV-INBC follows an approach than can be very limiting.

The probability of a document $\boldsymbol{d}_n$ in a DV-INBC model m is computed as

$$p(\boldsymbol{d}_n)_m = \sum_{j=1}^{m_M} p(\boldsymbol{d}_n|j) \cdot p(j), \tag{4.5}$$

where m_M is the number of clusters inside the m-th model, $p(j)$ is the prior probability of cluster j inside model m and $p(\boldsymbol{d}_n|j)$ is the j-th cluster likelihood. The computation of both values is explained in the following section.

4.3 The learning process

The learning process creates, deletes and updates clusters. For every new document d_n, its similarity to all current clusters is computed. As in INBC, if all clusters evaluate the current document as something new i.e. a *novelty*, a new cluster N is created with a Multinomial distribution over vocabulary V_n (found in document d_n) and represented by a centroid c_N, which is initially set to the bag-of-words representation of $\boldsymbol{d}_n$, as in (4.1). The values of the Multinomial distribution are computed following (4.4).

The criterion to evaluate a document as a novelty is based on the Jensen-Shannon distance (FUGLEDE; TOPSOE,)

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M), \tag{4.6}$$

where P and Q are two probability distributions over a common set of possible qualitative values and M is a *mean* distribution, defined as $M = \frac{(P+Q)}{2}$. The Jensen-Shannon distance is a symmetric version of the Kullback-Leibler divergence (D_{KL}) (KULLBACK; LEIBLER, 1951)

$$D_{KL}(P \parallel Q) = \sum_i P(i) \cdot ln \frac{P(i)}{Q(i)}, \tag{4.7}$$

which measures the information loss when using a distribution Q to represent data that was sampled from another distribution P.

The Kullback-Leibler measure is called a *divergence* because it is sensitive to the distribution taken as reference, which means that $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$. Moreover, its resulting values are contained in the interval $[0, \infty)$, which does not give a strong idea of how far the word probability distribution found in $\boldsymbol{d}_n$ is from a cluster j to decide whether $\boldsymbol{d}_n$ is a novelty or not. On the other hand, the Jensen-Shannon is a *symmetric divergence* (and therefore a *distance*) and has values contained on the interval $[0, \ln 2]$. Those two characteristics were the reason to use this distance as the basis for the novelty criterion in DV-INBC, allowing a better control of how close a document can be from a distribution to be assimilated by it.

To use the Jensen-Shannon distance, both P and Q have to be probability distributions over the same set of possible values. Since in DV-INBC clusters and documents can have different vocabularies, it is necessary to create a vocabulary to compute that distance. For this end, a fraction δ of the set of most frequent words inside cluster j, V_j^{top}, and the whole vocabulary found in document d_n, V_n, are used. Therefore, the common vocabulary used to compute the distance is defined as

$$V_j{}^c = V_n \cup V_j^{top}. \tag{4.8}$$

Besides, two Multinomial distributions should be made for the comparison: one to represent $\boldsymbol{d}_n$, $P_{d_n}^c$; and another to represent a cluster c_j, P_j^c. The probability of word w_i, $\forall w_i \in V_j{}^c$ in $P_{d_n}^c$ is computed as

$$p_{d_n}^c(w_i) = \frac{f_{d_n w_i} + 1}{\mid d_n \mid + \mid V_j{}^c \mid}, \tag{4.9}$$

where $|d_n| = \sum_{i=1}^{|V_j{}^c|} f_{d_n w_i}$. The equivalent for P_j^c is

$$p_j^c(w_i) = \frac{f_{j w_i} + 1}{\mid c_j^{top} \mid + \mid V_j{}^c \mid}, \tag{4.10}$$

where $\mid c_j^{top} \mid = \sum_{i=1}^{|V_j|} f_{j w_i}$. In this situation, it is necessary to perform a smoothing process over the values of all probabilities, since by using $V_j{}^c$ it is not guaranteed that both cluster j and document d_n contain the same set of words. For that end, the Laplacian smoothing technique was chosen, as can be seen in Equations 4.9 and 4.10.

Finally, using (4.8), (4.9) and (4.10) the novelty criterion of DV-INBC is applied by comparing the value of $D_{JS}(P_{d_n}^c \mid\mid P_j^c)$ to a fraction of the maximum value assumed by the D_{JS} distance, to decide whether $\boldsymbol{d}_n$ is a novelty or should be assimilated by an existing cluster. The novelty criterion is

$$D_{JS}(P_{d_n}^c \mid\mid P_j^c) > \tau \cdot \ln 2, \tag{4.11}$$

where τ is a hyperparameter. If at least one component evaluates the novelty criterion as False, no new component is created and the most similar cluster is updated using the words from $\boldsymbol{d}_n$.

By using $V_j{}^c$, the likelihood of the current document for a given cluster j is

$$p(\boldsymbol{d}_n|j) = \prod_{i=1}^{|V_n|} p_j^c(w_i)^{f_{nw_i}}. \tag{4.12}$$

After checking whether another cluster should be created, the model update process begins. Considering the most similar cluster (i.e. the one with the smallest D_{JS} distance) as s, the update process creates another vocabulary for that cluster, V_s^{new}, joining its current vocabulary V_s^{old} to the vocabulary from d_n, V_n

$$V_s^{new} = V_s^{old} \cup V_n. \tag{4.13}$$

Also, a new centroid $\boldsymbol{c}_s^{new}$ is created for s

$$\boldsymbol{c}_s^{new} = (f_{sw_1}^{new}, \cdots, f_{sw_i}^{new}, \cdots, f_{sw_{|V_j^{new}|}}^{new}), \tag{4.14}$$

with a dimensionality equal to $|V_s^{new}|$, adding more dimensions to the corresponding new words with their respective frequencies. The frequency of each word $f_{sw_i}^{new}$ is updated in the following manner: if a word already existing in cluster $\boldsymbol{c}_s$ occurred again in d_n, its value is incremented, otherwise, it is kept the same, as is shown below

$$f_{sw_i}^{new} = \begin{cases} f_{sw_i} + f_{d_n w_i}, & \text{if } w_i \in V_j^{old} \cap V_n \\ f_{d_n w_i}, & \text{if } w_i \in V_n \setminus V_j^{old} \\ f_{sw_i}, & \text{otherwise} \end{cases}. \tag{4.15}$$

Finally, the parameter vector $\boldsymbol{\theta}_s$ of the Multinomial distribution for component s is updated as well, by the following equation

$$\theta_{si}^{new} = \frac{f_{sw_i}^{new}}{\mid \boldsymbol{c}_s{}^{new} \mid}. \tag{4.16}$$

After checking the novelty of the current document or updating the current model, the posterior probability of each cluster j for $\boldsymbol{d}_n$ is computed by

$$p(j|\boldsymbol{d}_n) = \frac{p(\boldsymbol{d}_n|j)p(j)}{\sum_{j=1}^{N} p(\boldsymbol{d}_n|j)p(j)}, \tag{4.17}$$

and is used to update the sp accumulator of each component, which in turn is used to update the prior probability of each cluster $p(j)$ as in the INBC model. Also as in INBC, the sp accumulator is evaluated in conjunction to the age attribute of each component to decide which cluster(s) should be removed. Accumulators such as sps and spx do not exist in DV-INBC, since their usages were related to the update of Gaussian distributions parameters, which are not used in this model. The DV-INBC algorithm is presented in Algorithm 2.

4.4 The classification process

To categorize a classification document ${d_n}^{cl}$, a fraction δ of the most frequent words inside each cluster j, V_j^{top}, is used to create a classification vocabulary ${V_j}^{cl}$,

$${V_j}^{cl} = {V_n}^{cl} \cup V_j^{top}, \tag{4.18}$$

which is obtained by joining V_j^{top} with the vocabulary of document ${d_n}^{cl}$, ${V_n}^{cl}$. This is similar to the process to compute the similarity of documents and clusters used during training. The probability of each word w_i inside this vocabulary is then computed as

$$p_j{}^{cl}(w_i) = \frac{f_{jw_i} + 1}{\mid c_j \mid + \mid V_j^{cl} \mid}. \tag{4.19}$$

Using (4.5), the probability of a document for the m-th DV-INBC model $p(\boldsymbol{d}_{cl})_m$ is computed using (4.19) for the likelihood of each cluster. Then, to decide the class l of the document, the largest value among all is chosen, using the following equation:

$$l = \arg\max_{m \in M} p(\boldsymbol{d}_{cl})_m. \tag{4.20}$$

4.5 Conclusion

This chapter presented the DV-INBC algorithm, which extends the INBC method presented in Section 3. It maintains the main capabilities of INBC and changes the distribution used to model the data: from a Gaussian to a Multinomial distribution. The resulting model is, therefore, a mixture of Multinomial distributions.

The main differences between the two models are the usage of a single model per class, resulting in a set of DV-INBC models for a dataset. Also, since the probability distribution of each cluster has changed, the similarity criterion used also needed to be altered, and now the Jensen-Shannon distance is used. The other parts of the algorithm are still very similar to INBC. Also, since DV-INBC is based on INBC, a set of limitations were inherited, regarding the representation of topics and the process for generating documents.

On the following chapter, the performance of DV-INBC will be tested, using popular datasets and compared to recent methods, as well as popular ones.

Algorithm 2 DV-INBC

```
input: document $\boldsymbol{d}_n$
{Computes the similarity and likelihood of $\boldsymbol{d}_n$ to all components}
for $j = 1$ to $N$:
    $V_j^{top}$ = the $\delta \cdot |\boldsymbol{c}_j|$ top most frequent words from cluster j
    $V^c = V_n \cup V_j^{top}$
    $p^c_{d_n}(w_i) = \frac{f_{d_n w_i}+1}{|d_n|+|V^c|}, \forall w_i \in V^c$
    $p^c_j(w_i) = \frac{f_{j w_i}+1}{|c_j^{top}|+|V^c|}, \forall w_i \in V^c$
    $M = \frac{(P^c_{d_n}+P^c_j)}{2}$
    $D_{JS}(P^c_{d_n} \mid\mid P^c_j) = \frac{1}{2}D_{KL}(P^c_{d_n} \mid\mid M) + \frac{1}{2}D_{KL}(P^c_j \mid\mid M)$
    $p(\boldsymbol{d}_n|j) = \prod_{i=1}^{|V_n|} p^c_j(w_i)^{f_{nw_i}}$
end for
{Checks the novelty criterion for all clusters}
if $D_{JS}(P^c_{d_n} \mid\mid P^c_j) > \tau \cdot ln(2), \quad \forall j$:
    {If the novelty criterion is evaluated as true, creates a new component N}
    $c_N = \boldsymbol{d}_n$
    $V_N = V_n$
    $\theta_{Ni} = \frac{f_{d_n w_i}}{|c_N|}$
    $sp_N = 1$
    $p(N) = \frac{1}{\sum_{j=1}^{N} sp_j}$
    $age_N = 1$
end if
{Updates cluster with smallest $D_{JS}$ distance, according to (4.15)}
$V_s^{new} = V_s^{old} \cup V_n$
$\boldsymbol{c}_s^{new} = (f^{new}_{sw_1}, \cdots, f^{new}_{sw_i}, \cdots, f^{new}_{sw_{|V_s^{new}|}})$
$\theta^{new}_{si} = \frac{f^{new}_{sw_i}}{|\boldsymbol{c}_s{}^{new}|}$
{Computes the posterior probability of each cluster}
$p(j|\boldsymbol{d}_n) = \frac{p(\boldsymbol{d}_n|j)p(j)}{\sum_{j=1}^{N} p(\boldsymbol{d}_n|j)p(j)}, \quad \forall j$
{Updates the variables associated to all the clusters}
$sp_j = sp_j + p(j|\boldsymbol{d_n}), \forall j$
$p(j) = \frac{sp_j}{N}, \forall j$
$age_j = age_j + 1, \forall j$
{Checks if there are any spurious clusters}
for $j = 1$ to $N$:
    if $spurious_j = True$ :
        {Removes component j from the model}
    end if
end for
```

5 CLASSIFICATION OF DOCUMENTS USING DV-INBC

This chapter presents experiments using DV-INBC on popular datasets used for topic classification and sentiment analysis. The results obtained with DV-INBC are compared to other recent models and the performance of the model is compared to other popular and recent methods.

The chapter is divided in two sections: the first presents experiments on topic classification datasets and the last one the result obtained in a sentiment analysis task. For all experiments, the statistics for the values of the Jensen-Shannon divergence computed while using that dataset (for each model) are presented, to better understand the behaviour of this metric as well as to assess its usage in DV-INBC.

5.1 Classification of topics

This task is presented to evaluate the performance of DV-INBC when categorizing the main topic of text documents. Due to the classification method that is used by DV-INBC, in here three single-label datasets are presented: 20 Newsgroups[1], Reuters-R8 and WebKB[2].

5.1.1 20 Newsgroups

The 20 Newsgroups dataset is a collection of approximately 20,000 posts taken from Usenet groups. There are 20 classes in this dataset, representing different topics, almost evenly separated. Table 5.1 shows the different existing classes in the dataset, and how many documents there are in each one.

The 20 Newsgroups dataset is available in several formats. In this thesis, the *by-date* Matlab/Octave version[3] (which has 11,269 files in the training set and 7,505 files in the test one) was used. The *by-date* version means that the dataset was split by ordering all instances by the time of posting, making a training set composed of examples that precede all the ones found in the test set. This version was used to compare the results of DV-INBC to other recent approaches.

The age_{min} = 10 and sp_{min} = 2.5 parameters were fixed, and τ and δ were found by grid search, using a validation set composed by 5% of the training set, i.e from the original training set composed of 11,269 documents, a fraction of 5% was taken to create a validation set, on which parameter estimation was conducted. During validation, it was observed that for lower values of the τ parameter, the performance of the model was much worse than for larger ones (achieving accuracy values smaller than 5%). For each particular value of τ, changes in the δ parameter had a different impact: when considering the lower values of τ, the model performed better with extreme values of δ, while values located in the middle region made a worse perfor-

[1] The 20 Newsgroups dataset can be downloaded from http://qwone.com/~jason/20Newsgroups/

[2] The Reuters-R8 and WebKB datasets can be downloaded from http://web.ist.utl.pt/acardoso/datasets/

[3] This version can be downloaded from http://qwone.com/~jason/20Newsgroups/20news-bydate-matlab.tgz

Table 5.1: The 20 Newsgroups dataset documents per class in the original training set.

Class name	**Number of documents**
comp.graphics	480
comp.os.ms-windows.misc	581
comp.sys.ibm.pc.hardware	572
comp.sys.mac.hardware	587
comp.windows.x	575
rec.autos	592
rec.motorcycles	582
rec.sport.baseball	592
rec.sport.hockey	596
sci.crypt	594
sci.electronics	598
sci.med	594
sci.space	591
misc.forsale	594
talk.politics.misc	593
talk.politics.guns	599
talk.politics.mideast	545
talk.religion.misc	564
alt.atheism	464
soc.religion.christian	376

mance. Still, the results were very poor. Those values are presented in Table 5.2, as a function of each parameter combination.

Table 5.2: The error rate of DV-INBC on the 20 Newsgroups dataset during the validation phase (%).

τ \ δ	0.25	0.5	0.75	1.0
0.25	75.7	93.63	96.63	51.59
0.5	12.76	12.23	11.87	**11.52**
0.75	12.76	12.23	11.87	**11.52**

For $\tau \geq 0.5$, the performance increased as δ was increased as well, achieving the best result with $\delta = 1.0$. In that situation, both $\tau = 0.5$ and $\tau = 0.75$ yielded the same and best results for validation.

Finally, after performing the grid search, it was decided to choose randomly between the two mentioned best results to train DV-INBC with the full training set. The parameter set chosen was $\tau = 0.5$ and $\delta = 1.0$. Using this set of parameters, DV-INBC achieved an error rate of 21.61% over the 20 Newsgroups dataset. The amount of clusters inside each DV-INBC model was the same and equal to one.

Table 5.3 presents a comparison with other recent results. The work from (LAROCHELLE et al., 2012) adapts a Restricted Boltzman Machine (RBM) (SMOLENSKY, 1986)(HINTON, 2002) to perform classification and assesses its performance on several tasks, among which is document classification using the 20 Newsgroups dataset. Essentially, an RBM is an undirected graphical model with binary variables. To better use an RBM for classification, a hybrid training objective was created mixing both generative and discriminative objectives. To represent each document, a fixed vocabulary composed by the 5,000 most frequent words of the training part of the dataset was used. The method presented in (DAUPHIN; BENGIO, 2013) uses an RBM as well, following a two-phase training: first the RBM is trained and then the adjusted model is used to initialize the hidden layer of a Multilayer Perceptron (MLP) (BISHOP, 2006) neural network, which is finally used to perform the document classification task. In this approach, a fixed vocabulary composed by all the 61,188 words from the training dataset was used. Both works use a binary representation for documents, considering only the occurence of words and not their frequency.

Table 5.3: Comparison of results on the 20 Newsgroups dataset.

Method	**Error rate** (%)
ClassRBM (Larochelle et al. 2012)	**23.8**
RBM-MLP (Dauphin and Bengio 2013)	**20.5**
DV-INBC	**21.61**

The Jensen-Shannon divergence values computed for each document for all classes of the dataset (during the whole training) is shown in Figures 5.1 - 5.4. On all images, it can be noted that the value of the distance increases as more words were added to the vocabulary of the only cluster instantiated inside each model. This is expected since by increasing the vocabulary of the cluster, it increases the chance of becoming more different for future documents. Therefore, the value of the distance is higher as more documents are used. The mean value and the standard deviation of the Jensen-Shannon distance for each class is shown in Table 5.4.

5.1.2 Reuters-R8

The Reuters-R8 dataset is a modified version of the traditional Reuters-21578 dataset, which is composed by newswire articles from Reuters agency, which were manually classified into many categories by personel from that company and had multi-label instances. The modified R8 version was originally presented in (CARDOSO-CACHOPO, 2007), and was created by

Table 5.4: Mean value and standard deviation of the Jensen-Shannon distance computed for each class in the 20 Newsgroups dataset

Class	**Mean value**
comp.graphics	0.22 ± 0.058
comp.os.ms-windows.misc	0.19 ± 0.057
comp.sys.ibm.pc.hardware	0.18 ± 0.044
comp.sys.mac.hardware	0.21 ± 0.048
comp.windows.x	0.18 ± 0.05
rec.autos	0.24 ± 0.028
rec.motorcycles	0.14 ± 0.04
rec.sport.baseball	0.19 ± 0.052
rec.sport.hockey	0.19 ± 0.039
sci.crypt	0.2 ± 0.047
sci.electronics	0.2 ± 0.051
sci.med	0.25 ± 0.043
sci.space	0.19 ± 0.052
misc.forsale	0.19 ± 0.051
talk.politics.misc	0.20 ± 0.038
talk.politics.guns	0.24 ± 0.057
talk.politics.mideast	0.21 ± 0.055
talk.religion.misc	0.24 ± 0.06
alt.atheism	0.22 ± 0.059
soc.religion.christian	0.20 ± 0.052

only considering documents with a single topic and the classes which still had at least one train and one test example after that modification. Table 5.5 presents all the categories present in Reuters-R8 dataset with the amount of documents for each class.

To evaluate DV-INBC, among the many versions available for this dataset on its website, the *all-terms* version was the chosen one. In this version, as stated on the page of the dataset, the following pre-processing was applied:

- all TAB, NEWLINE and RETURN characters were substituted by SPACE;
- only letters were kept, and all other characters were turn into SPACE;
- all letters were turn to lowercase;
- multiple spaces were substituted by a single one;
- the title/subject of the document is simply added in the beggining of the text of the document.

A validation phase was performed to find the best parameters to this dataset, and for that a random subset composed by 60% of all training documents was used, i.e. it was created another

Table 5.5: The Reuters-R8 dataset classes.

Class name	**Number of documents**
earn	2,840
aqc	1,596
trade	251
ship	108
grain	41
crude	253
interest	190
money-fx	206

set for parameter estimation, composed by 60% of the original training set, and parameter estimation was tested on the remaining 40%. The experiments have shown that lower values of τ yielded worse results, with an unstable behavior on that situation. For $\tau \geq 0.5$ a better performance was achieved, which was improved as the value of δ was also increased. This behavior is presented in Table 5.6.

Table 5.6: The error rate of DV-INBC on the Reuters-R8 dataset, during the validation phase (%).

τ \ δ	0.25	0.5	0.75	1.0
0.25	95.26	63.73	95.31	94.63
0.5	44.9	36.12	31.66	**27.2**
0.75	44.9	36.12	31.66	**27.2**

The best performance was obtained with two parameter sets: $\tau = 0.5$ and $\delta = 1.0$ and $\tau = 0.75$ with $\delta = 1.0$. To select the set of parameters for the final evaluation, a random choice was made among all the parameter sets that gave the best performance, and the chosen set was $\tau = 0.75$ and $\delta = 1.0$. Finally, using this set of parameters, the final models had an accuracy level of 79.3% and each one had a unique cluster inside. Other algorithms were tested on this dataset to compare how DV-INBC performs against other popular (although not recent) approaches, and their results are presented in Table 5.7.

Table 5.8 presents the mean values of the Jensen-Shannon divergence for this experiments, and Figures 5.5 and 5.6 shows the values of the divergence during training, for all classes of the dataset. As in the previous experiment, the behavior of the distance is to increase as more documents are used for training, which makes the vocabulary of the clusters more different from the ones of each document. This again is an expected behavior and tends to increase the value

Table 5.7: Comparison of results on the Reuters-R8 dataset.

Method	**Accuracy** (%)
SVM (Linear kernel)	**94.93**
Naive Bayes	**95.39**
DV-INBC	**79.3**

of the distance.

Table 5.8: Mean value and standard deviation of the Jensen-Shannon distance computed for each class in the Reuters-R8 dataset

Class	**Mean value**
earn	0.31 ± 0.058
acq	0.26 ± 0.053
trade	0.17 ± 0.051
ship	0.11 ± 0.033
grain	0.86 ± 0.028
crude	0.17 ± 0.048
interest	0.16 ± 0.045
money-fx	0.15 ± 0.048

5.1.3 WebKB

This dataset is composed by webpages collected by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. The webpages were originally taken from computer science departments of various universities in 1997 and manually classified into seven classes, which indicates the type of content for that page: student, faculty, staff, department, course, project, and other. For each class, the collection contains pages from four universities: Cornell, Texas, Washington, Wisconsin and other many pages collected from other universities.

As informed on the page of the dataset, the version that was used on this thesis (called *stemmed*) had three classes removed (department, staff and other), and underwent the same pre-processing that was used on the Reuters-R8 dataset (as explained in 5.1.4), plus removal of words less than three characters long, stopwords and the application of a stemmer. Finally, before conducting the experiments of this thesis, it was noted that the dataset still had empty documents in the training set, which were then removed. Table 5.9 shows the final set of classes in the dataset with the number of documents in each one.

In the validation phase, following the same approach of the previous experiment, a random

Table 5.9: The WebKB dataset classes.

Class name	Number of documents
student	1085
faculty	745
project	335
course	620

subset 60% of the original training set was used for training, and the remaining documents were used for validation. To find the best parameter set for all the models, a grid search was applied. Since this dataset was heavily pre-processed, the data had less noise, and therefore the performance of DV-INBC was much better, even in the worst situations. The model had its best performance with parameters $\tau = 0.25$ and $\delta = 0.75$. It was noted that larger values for τ yielded a lower performance for DV-INBC, when observing the value of δ on which its performance was the best one, a behavior that was different from the ones observed in the other experiments, when by increasing the value of τ the result was always improved or stayed the same. This can be explained by the increased quality of the pre-processed data, which now has a more different vocabulary for each subtopic in each class. Besides, the vocabulary is more informative of the document class due to the stopword removal and stemming processes that were applied. The values achieved during the validation phase are shown in Table 5.10.

Table 5.10: The error rate of DV-INBC on the WebKB dataset, during the validation phase (%).

τ \ δ	0.25	0.5	0.75	1.0
0.25	19.74	27.01	16.24	29.53
0.5	19.74	18.76	18.49	17.95
0.75	19.74	18.76	18.49	17.95

Finally, the best parameter set of the previous phase was used to perform the final evaluation of DV-INBC over the dataset, and the accuracy level achieved was 83.87%. As in the previous experiment, other popular algorithms were applied to compare the performance of DV-INBC. In this dataset, the number of clusters inside each model was different, and Table 5.11 shows how many clusters each model created. Finally, Table 5.12 shows the results obtained by DV-INBC and by the other methods on this dataset.

Figure 5.7 shows the mean values of the Jensen-Shannon divergence and the standard deviation of the distance obtained during training, for all classes of the dataset. Table 5.13 shows the mean value and standard deviation of the divergence inside each class, computed after training.

Table 5.11: Clusters by class created in the WebKB dataset.

Class	**Number of clusters**
student	**3**
faculty	**3**
project	**2**
course	**4**

Table 5.12: Comparison of results on the WebKB dataset.

Method	**Accuracy** (%)
SVM (Linear kernel)	**86.98**
Naive Bayes	**84.02**
DV-INBC	**83.87**

Figure 5.7 shows that during training the clusters created tend to get similar causing an increase in the mean value of the distance (a similar behavior was observed in the other experiments, but in that situation only one cluster was created). When a new cluster is created, the Jensen-Shannon distance achieves a smaller value, and as the training continues, its mean value continues to grow. This is also observed by analysing the value of the standard deviation of the distance: it peaks everytime a cluster is created, indicating that, at that moment, the difference among the values of the distance was higher; and gets smaller the moments between creation of clusters. This shows that when a cluster is created, its distance to the current document is much smaller when compared to the other clusters, resulting in a higher standard deviation and a smaller value for the mean.

Table 5.13: Mean value and standard deviation of the Jensen-Shannon distance computed for each class in the Reuters-R8 dataset

Class	**Mean value**
student	0.14 ± 0.026
faculty	0.14 ± 0.024
project	0.13 ± 0.027
course	0.14 ± 0.023

5.1.4 Evaluating the use of priors on the classification

After performing the topic classification experiments, it was noted the poor performance of DV-INBC on the Reuters-R8 dataset, an experiment where the method had a much worse performance than a traditional Naive Bayes algorithm. A major difference between those two models is the use of prior probabilities in the classification, which helps improve performance in datasets where there is a large difference among the number of documents in each class.

To evaluate how this information impacts on the performance of DV-INBC, the best models for all datasets used in the topic classification tasks were tested again, now using the prior probabilities of each class. For this, a small change to the classification process of DV-INBC was necessary, yielding the following:

$$l = \arg\max_{m \in M} p(m) \cdot p(\boldsymbol{d}_{cl})_m, \tag{5.1}$$

where $p(m)$ is the prior probability of the class represented by the m-th DV-INBC model, computed as

$$p(m) = \frac{\mid \mathcal{D}_m \mid}{\mid \mathcal{D} \mid}, \tag{5.2}$$

where $\mathcal{D}_m$ is the set of all documents of class m in the training set and $\mathcal{D}$ is the set of all training documents.

Table 5.14 show the results of using prior probabilities for each class on the datasets.

Table 5.14: The effect of class priors in the categorization performance of DV-INBC

Datset	**Accuracy without priors (%)**	**Accuracy with priors (%)**
Reuters-R8	**79.3**	**87.8**
WebKB	**83.87**	**84.16**
20 Newsgroups	**78.39**	**78.39**

As could be observed, the performance of DV-INBC on the Reuters-R8 dataset had a larger improvement than in the other ones, increased by 8.5%, which is still 7.59% lower than the performance of a Naive Bayes algorithm on the same dataset. In the other datasets, the improvement was much smaller or even null. This is due to the other datasets having a more equal distribution of classes, with the less skewed distribution being the one of the 20 Newsgroups dataset, where the improvement was null.

5.2 Sentiment classification

For this task, the IMDB dataset was used. It was initially presented in (MAAS et al., 2011) and is a large repository of highly polar reviews about movies[4]. It is composed by a training set of 25,000 documents and a test set with the same amount of documents. There are only two classes, *good* and *bad*.

The IMDB dataset is composed by reviews taken from the Internet Movie Database website[5], where users post analysis and opinions from TV shows (e.g. movies, series, etc). When analysing the sentiment of the review (i.e. its polarity) a simple analysis of a bag of unigrams may not give a good performance, although a better result can be achieved by using bigrams, as shown in (WANG; MANNING, 2012). This means that when classifying movie reviews based on a bag-of-words model, while using unigrams, it is expected to have a worse performance than other approaches that use bigrams or that can utilize the word order to have a better semantic representation, as the technique presented in (JOHNSON; ZHANG, 2014). For instance, although the two following sentences have very similar bag-of-words vectors, they completely differ in meaning: *"I like this movie"* and *"I didn't like this movie"*.

Table 5.15 shows two random items of the IMDB dataset, one from each class and, as can be seen, the document size can vary greatly, as well as the type of language and vocabulary used on each class. Also, as mentioned before, a simple analysis of the vocabulary used in the review does not make clear its polarity.

Regarding the training of DV-INBC, the two following parameters were used with values $age_{min} = 10$ and $sp_{min} = 2.5$. Also, a grid search was performed to find the best values for τ and δ parameters. For that reason, a smaller set was used for training (with 2,500 documents per class) and an also smaller one for validation (with 1,250 documents per class). Both subsets were randomly created from the original training set.

The difference between the best and worst results in the validation phase was small when compared to the datasets used in the document classification task, with accuracy values ranging from around 50% to 81%. However, the behavior of DV-INBC was more unstable in this dataset. The best results were obtained with two different combinations of parameter values: $\tau = 0.75$ and $\delta = 0.25$ and $\tau = 0.5$ with $\delta = 0.25$. For all values of τ, extreme values of δ gave the same or better results than values located in the middle region. The values from the validation phase are shown in Table 5.16.

After performing the validation phase, the set of parameters used to train the model using the full training set was chosen randomly from the parameters that yielded the same result in that phase. The parameter set chosen was $\tau = 0.75$ and $\delta = 0.25$, and the result obtained was an error rate of 18.53%. Each DV-INBC model created only one cluster.

Table 5.17 compares the results from this work to others reported by (LE; MIKOLOV,

[4] The dataset can be downloaded from http://ai.stanford.edu/~amaas/data/sentiment/

[5] http://www.imdb.com/

Table 5.15: Two random exaples from the IMDB dataset

Example of a positive review
I'm a male, not given to women's movies, but this is really a well done special story. I have no personal love for Jane Fonda as a person but she does one Hell of a fine job, while DeNiro is his usual superb self. Everything is so well done: acting, directing, visuals, settings, photography, casting. If you can enjoy a story of real people and real love - this is a winner.
Example of a negative review
or anyone who was praying for the sight of Al Cliver wrestling a naked, 7ft tall black guy into a full nelson, your film has arrived! Film starlet Laura Crawford (Ursula Buchfellner) is kidnapped by a group who demand the ransom of $6 million to be delivered to their island hideaway. What they don't count on is rugged Vietnam vet Peter Weston (Cliver) being hired by a film producer to save the girl. And what they really didn't count on was a local tribe that likes to offer up young women to their monster cannibal god with bloodshot bug eyes.<br /><br />Pretty much the same filming set up as CANNIBALS, this one fares a bit better when it comes to entertainment value, thanks mostly a hilarious dub track and the impossibly goofy monster with the bulging eyes (Franco confirms they were split ping pong balls on the disc's interview). Franco gets a strong EuroCult supporting cast including Gisela Hahn (CONTAMINATION) and Werner Pochath (whose death is one of the most head-scratching things I ever seen as a guy who is totally not him is shown - in close up - trying to be him). The film features tons of nudity and the gore (Tempra paint variety) is there. The highlight for me was the world's slowly fistfight between Cliver and Antonio de Cabo in the splashing waves. Sadly, ol' Jess pads this one out to an astonishing (and, at times, agonizing) 1 hour and 40 minutes when it should have run 80 minutes tops. <br /><br />For the most part, the Severin DVD looks pretty nice but there are some odd ghosting images going on during some of the darker scenes. Also, one long section of dialog is in Spanish with no subs (they are an option, but only when you listen to the French track). Franco gives a nice 16- minute interview about the film and has much more pleasant things to say about Buchfellner than his CANNIBALS star Sabrina Siani.

Table 5.16: The error rate of DV-INBC on the IMDB dataset, during the validation phase (%).

δ τ	0.25	0.5	0.75	1.0
0.25	21.48	49.68	49.36	23.72
0.5	18.08	49.32	21.84	20.72
0.75	18.08	18.32	18.16	18.16

2014), which, at the time of writing this text, are the best published ones. Also the same table shows results from other methods. The work from (MAAS et al., 2011) uses an SVM trained with inputs represented by LDA features, with words as vectors (by using the relation of each one to each topic in the collection). The approach used in (WANG; MANNING, 2012) also uses an SVM, trained with bag of bigrams, and shows that it is better to use bigrams to perform sentiment analysis. Also, it shows that for smaller texts, probabilistic methods such as the Multinomial Naive Bayes can be better than discriminative ones, as an SVM. However, the same can not be said in the case of longer texts, a scenario where an SVM has a better performance. The *seq2-bown-CNN* method (JOHNSON; ZHANG, 2014) uses a Convolutional Neural Network (CNN) (LECUN et al.,) with two sequential convolutional layers and another convolutional layer with a bag-of-words input, representing the entire document. Finally, the Paragraph Vector technique presented in (LE; MIKOLOV, 2014) uses a different representation, using vector representation for words that are semantically related, i.e. vectors representing similar words are closer than those which represent words with very different meanings. It is a semi-supervised technique that learns vector representations of variable pieces of texts, such as sentences and documents, and uses those to train a logistic regression for the classification task.

Table 5.17: Comparison of results on the IMDB dataset.

Method	**Error rate** (%)
SVM (with LDA features) (Maas et al. 2011)	**32.58**
NBSVM-bi (Wang and Manning 2012)	**8.72**
seq2-bown-CNN (Johnson and Zhang 2014)	**7.67**
Paragraph Vector (Le and Mikolov 2014)	**7.42**
DV-INBC	**18.53**

Figure 5.8 shows the mean values of the Jensen-Shannon divergence and the standard deviation of the distance obtained during training, for all classes of the dataset. As in the other experiments where only one cluster was created by class, it can be seen that the mean value of the distance always increases, as more words are added to the vocabulary of each cluster. Table 5.18 shows the mean value and standard deviation of the Jensen-Shannon divergence inside each class, computed after training.

Table 5.18: Mean value and standard deviation of the Jensen-Shannon distance computed for each class in the IMDB dataset

Class	**Mean value**
positive	0.31 ± 0.04
negative	0.32 ± 0.04

5.3 Conclusions

This chapter presented an assessment of the performance of DV-INBC on two different tasks: topic classification and sentiment analysis. The first was the main focus of the model and therefore three experiments were made, while in the last task only one test was conducted. For topic classification, the datasets were 20 Newsgroups, Reuters-R8 and WebKB, and for sentiment analysis the chosen dataset was IMDB.

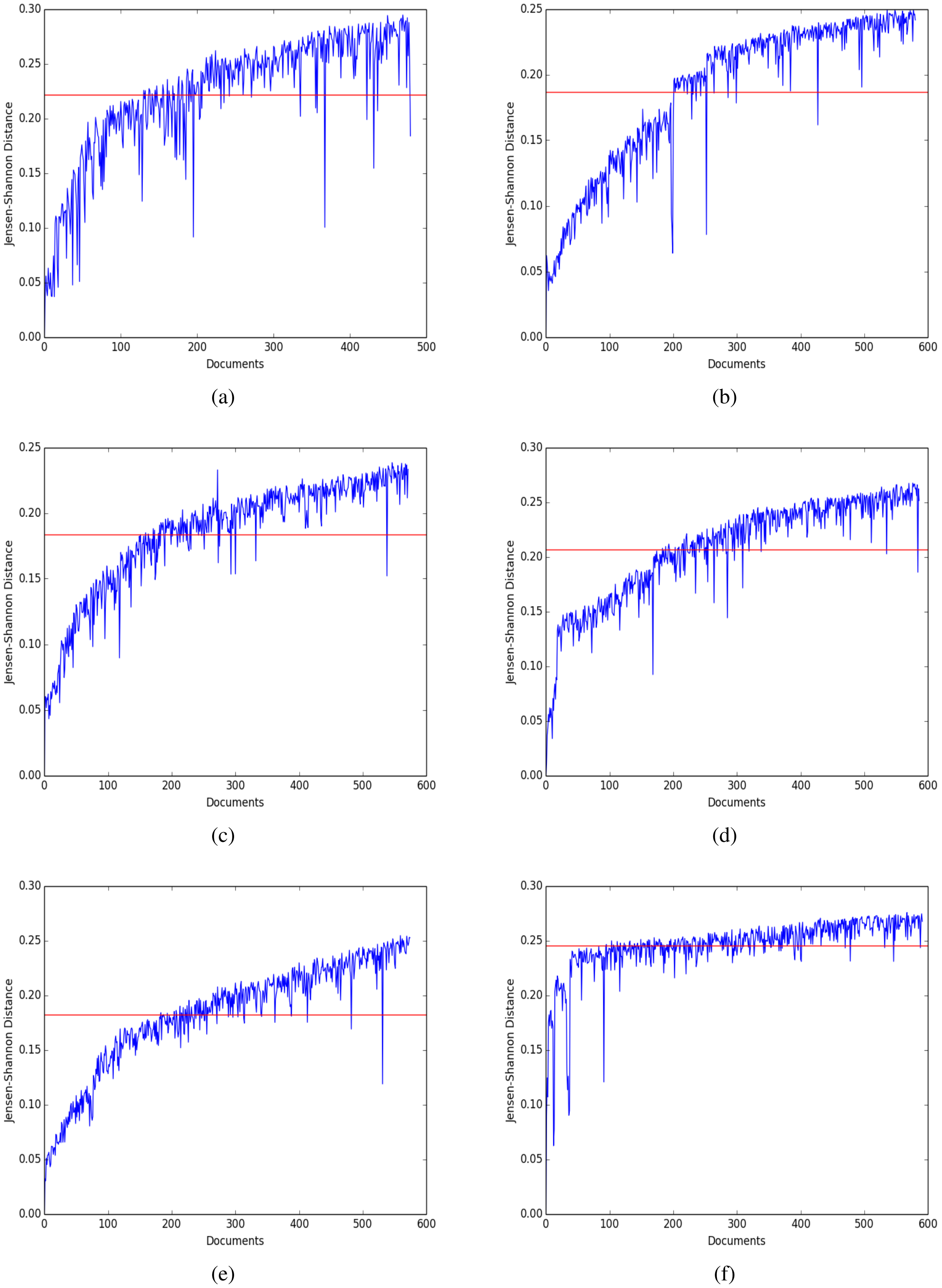
It was observed that the model had competitive performance to the state of the art on the 20 Newsgroups dataset, and regular to worse performances on two other topic classification datasets. On the IMDB dataset, the performance was much worse than the best recent result that it was compared to. This behavior can be explained by the fact that the the task of sentiment analysis depends of the word order on each document, since the semantics of the text strongly dependends on it. On the other hand, the 20 Newsgroups dataset is commonly used in topic classification tasks, where word order is not very important to achieve a good performance, and a simple bag-of-words model is enough in most of cases. Since DV-INBC uses the bag-of-words model to represent documents, it was expected a worse performance on the IMDB dataset. However it is worth mentioning that the error rate achieved by DV-INBC was smaller than the one of the method used in (MAAS et al., 2011), which used features extracted from an LDA model to describe each document. This difference of performances between the two methods can be explained by the fact that DV-INBC uses a separate model for each class, which allows it to better represent each one, while LDA uses a single model for all classes, representing them using the topic space found by the algorithm. The two other methods presented in Table 5.17 either maintain word order or use bigrams to represent each text, two approaches that can improve performance.

On the other two datasets, Reuters-R8 and WebKB, the performance of DV-INBC was compared to two popular algorithms, SVM and Naive Bayes, and it could be observed a great difference of performances from the first dataset to the last. In Reuters-R8, the performance of DV-INBC was worse than an SVM and a Naive Bayes. This can be explained by the fact that a Naive Bayes method uses a complete vocabulary to describe each class, using even the words that did not occur in that class. However, DV-INBC uses only an approximation of that vocabulary, which is improved at every new document by incrementing the vocabulary of each model. This means that, at any time, if a union of the vocabularies of each cluster inside a model was made, an approximation of the complete vocabulary of that class would be obtained. Also, the Reuters-R8 dataset has a very skewed class distribution, with many more documents in only one class than in all the others. Since Naive Bayes uses this information jointly with the full vocabulary of each class, by modeling it as prior probabilities, it is expected to have a better performance in that situation. This was proved by repeating the experiments using that information during classification.

Finally, in the WebKB dataset DV-INBC had an equivalent performance to a Naive Bayes algorithm, achieving almost the same accuracy value. However, it is worth mentioning that DV-INBC used less information for that and also created clusters inside each model that help separate the different subtopics related to each main topic.

The experiments have also shown that the two main parameters of DV-INBC have very different impacts on the performance of the model. The τ parameter seems to have a higher effect on the performance, changing indiretly the quantity of clusters created during training. If it is set to a small value, many clusters are created and otherwise a smaller number of clusters (or even a single one) are instantiated.

On the other hand, the δ parameter was ineffective or had an unstable behavior for some values of τ, suggesting that its influence on DV-INBC should be revised or maybe the parameter should be exchanged by another one. This parameter was introduced to observe how the feature selection performed by DV-INBC when comparing the similarity of a cluster and a document could impact on the performance of the model, and it was expected to find a subset of the most frequent words inside each cluster that could give a good comparison. As the experiments have presented, the quality of the comparison has a stronger relation to the quality of the data and not with the amount of words used in the comparison. When tested against no pre-processed datasets (as was the case with 20 Newsgroups and Reuters-R8), the best results were obtained with a single cluster per model and using the entire vocabulary of each cluster for the comparison. This behavior can be explained by the fact that in those datasets there are still stopwords in the documents and therefore more words are needed to understand the real topic of each text, which tend to be less frequent. This means that a larger portion of the vocabulary is needed for the comparison. This behavior was not observed in the WebKB dataset, due to the pre-processing applied. Therefore, a smaller portion of the vocabulary could be used to achieve a good result. Nevertheless, the results show that a selection of the most frequent words is not robust enough to be further used, and so this part of DV-INBC should to be redesigned.



(a)

(b)

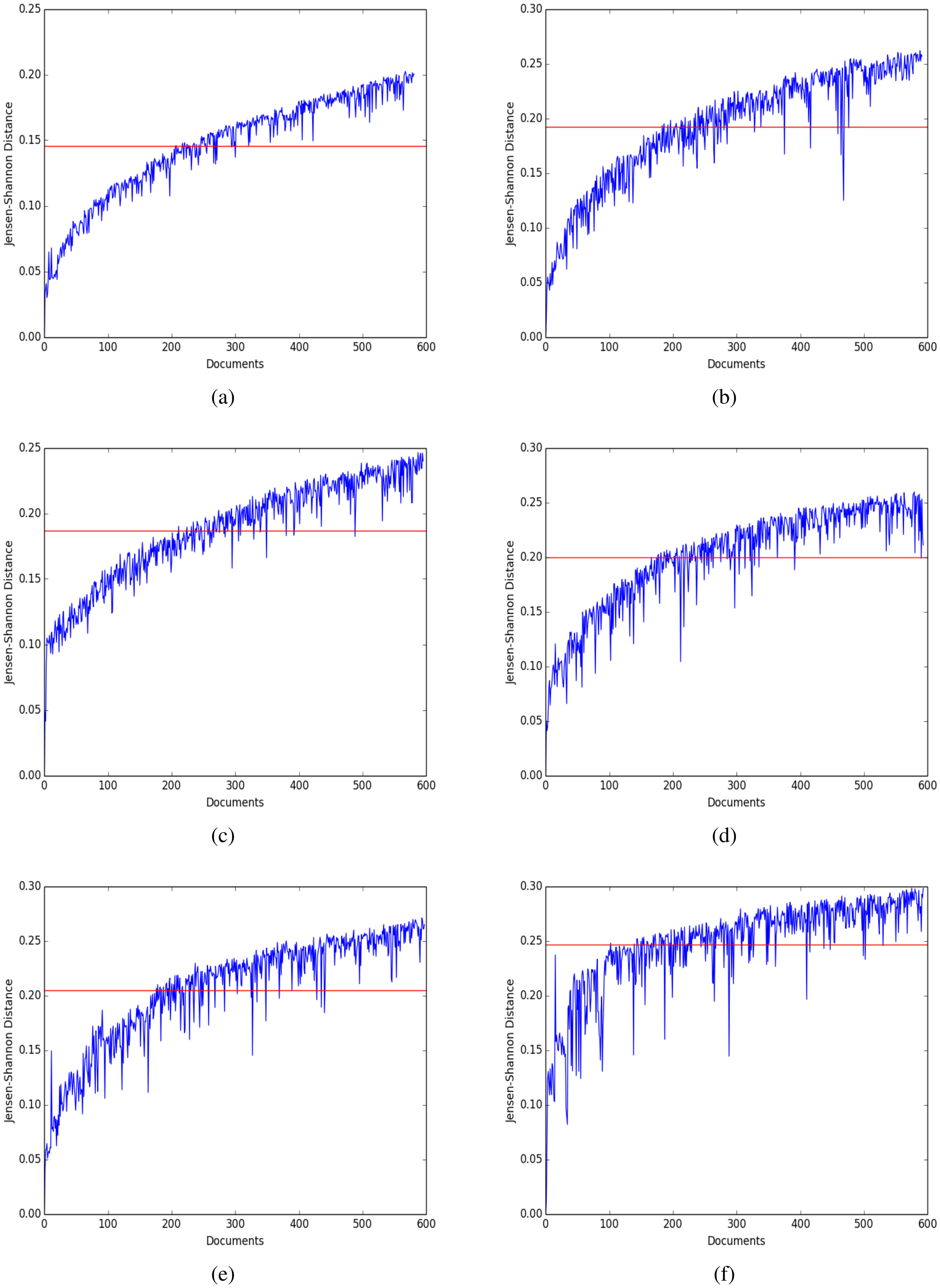
(c)

(d)

(e)

(f)

Figure 5.1: The mean value of the Jensen-Shannon distance, computed for each document during training on the 20 Newsgroups dataset, for classes (a) *comp.graphics*, (b) *comp.os.ms-windows.misc*, (c) *comp.sys.ibm.pc.hardware*, (d) *comp.sys.mac.hardware*, (e) *comp.windows.x* and (f) *rec.autos*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



(a)

(b)

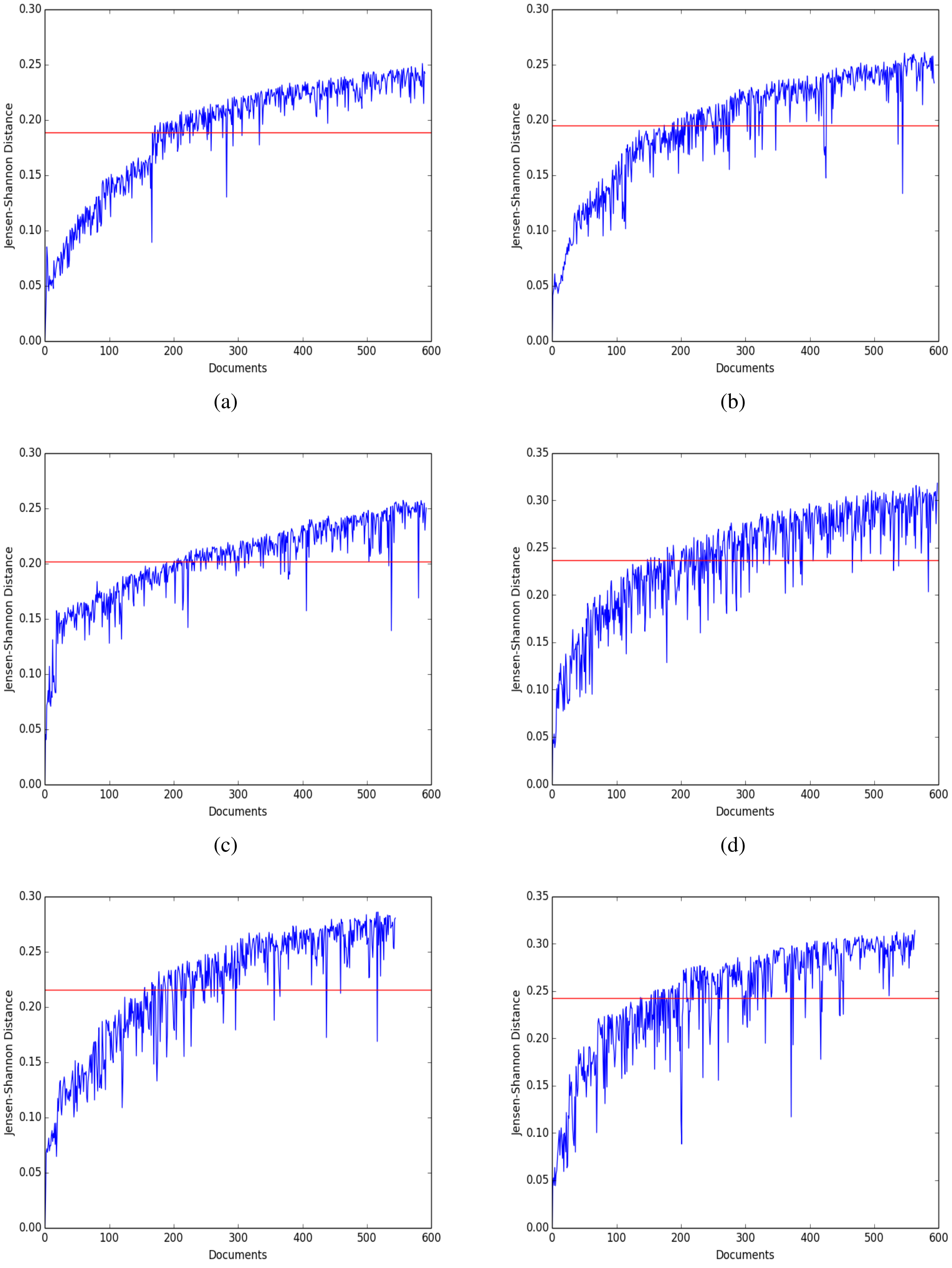
(c)

(d)

(e)

(f)

Figure 5.2: The mean value of the Jensen-Shannon distance, computed for each document during training on the 20 Newsgroups dataset, for classes (a) *rec.motorcycles*, (b) *rec.sport.baseball*, (c) *rec.sport.hockey*, (d) *sci.crypt*, (e) *sci.electronics* and (f) *sci.med*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



(a)

(b)

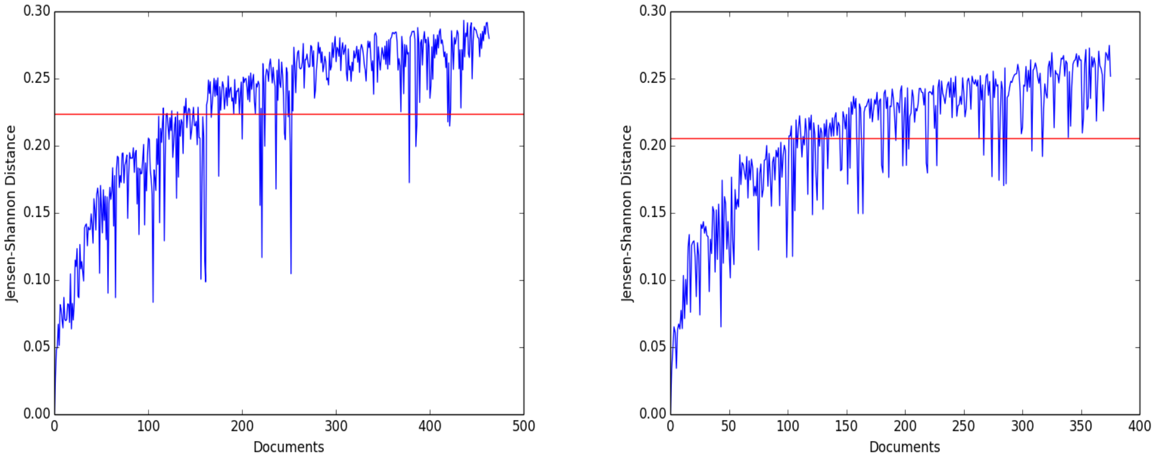
(c)

(d)

(e)

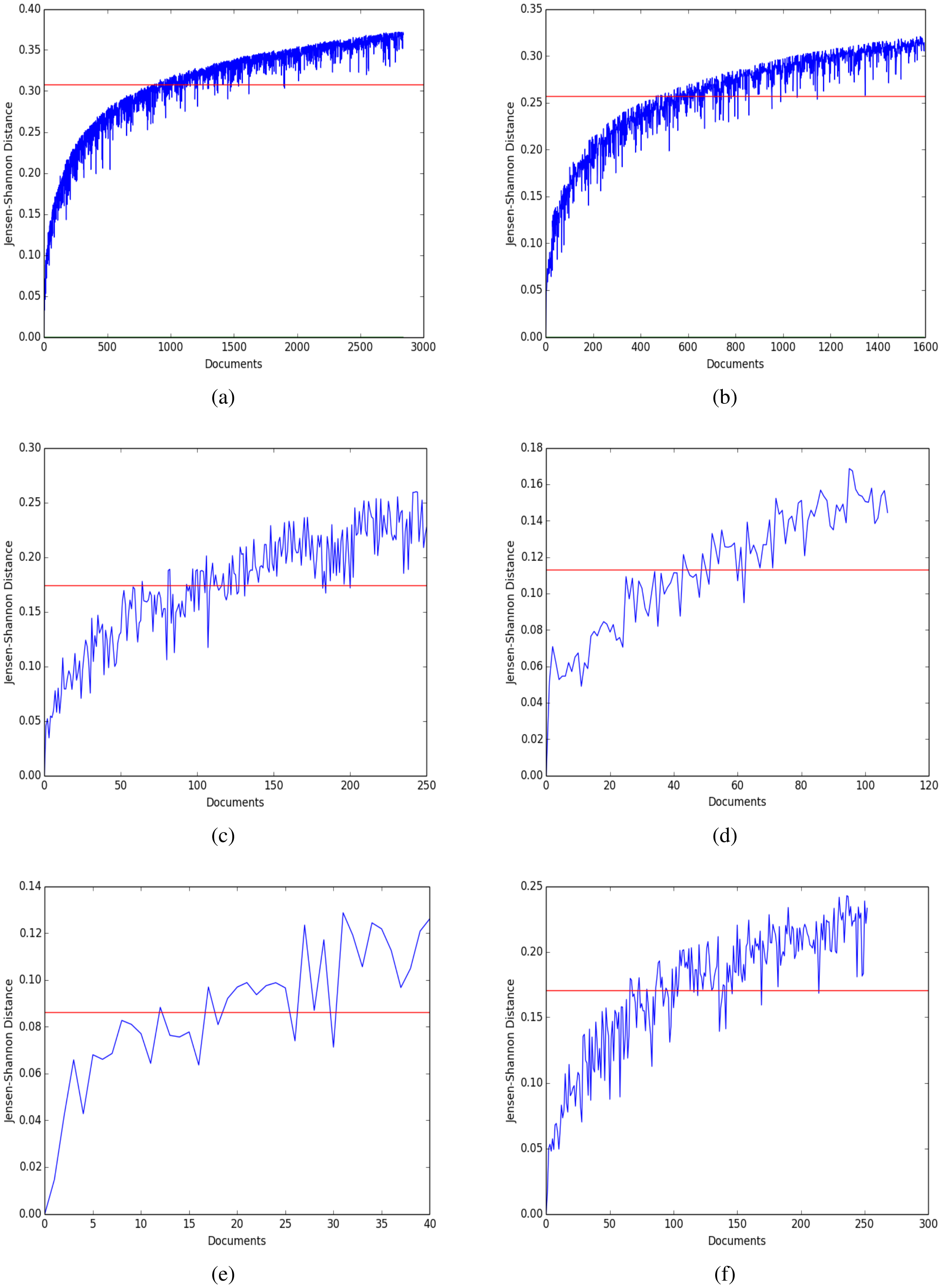
(f)

Figure 5.3: The mean value of the Jensen-Shannon distance, computed for each document during training on the 20 Newsgroups dataset, for classes (a) *sci.space*, (b) *misc.forsale*, (c) *talk.politics.misc*, (d) *talk.politics.guns*, (e) *talk.politics.mideast* and (f) *talk.religion.misc*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



(a) (b)

Figure 5.4: The mean value of the Jensen-Shannon distance, computed for each document during training on the 20 Newsgroups dataset, for classes (a) *alt.atheism* and (b) *soc.religion.christian*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



(a)

(b)

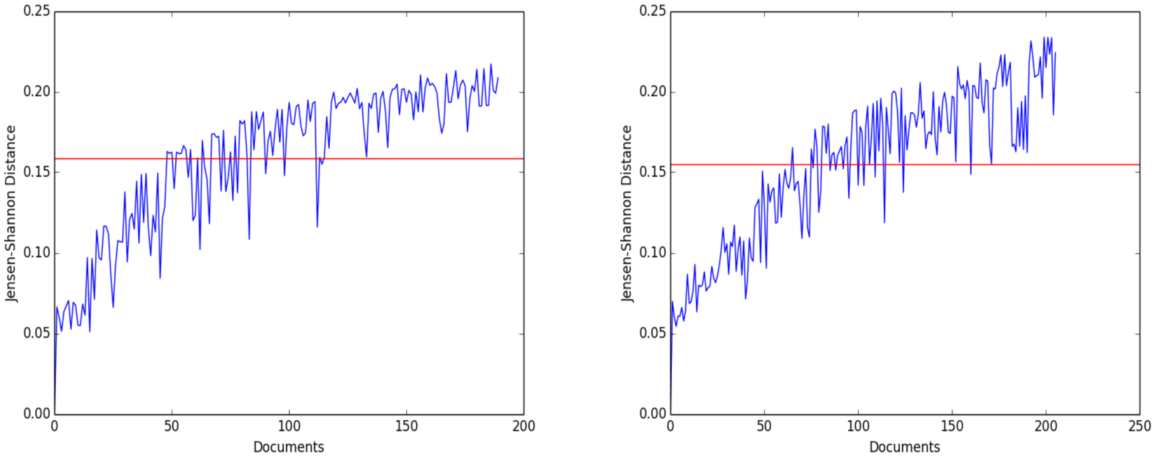
(c)

(d)

(e)

(f)

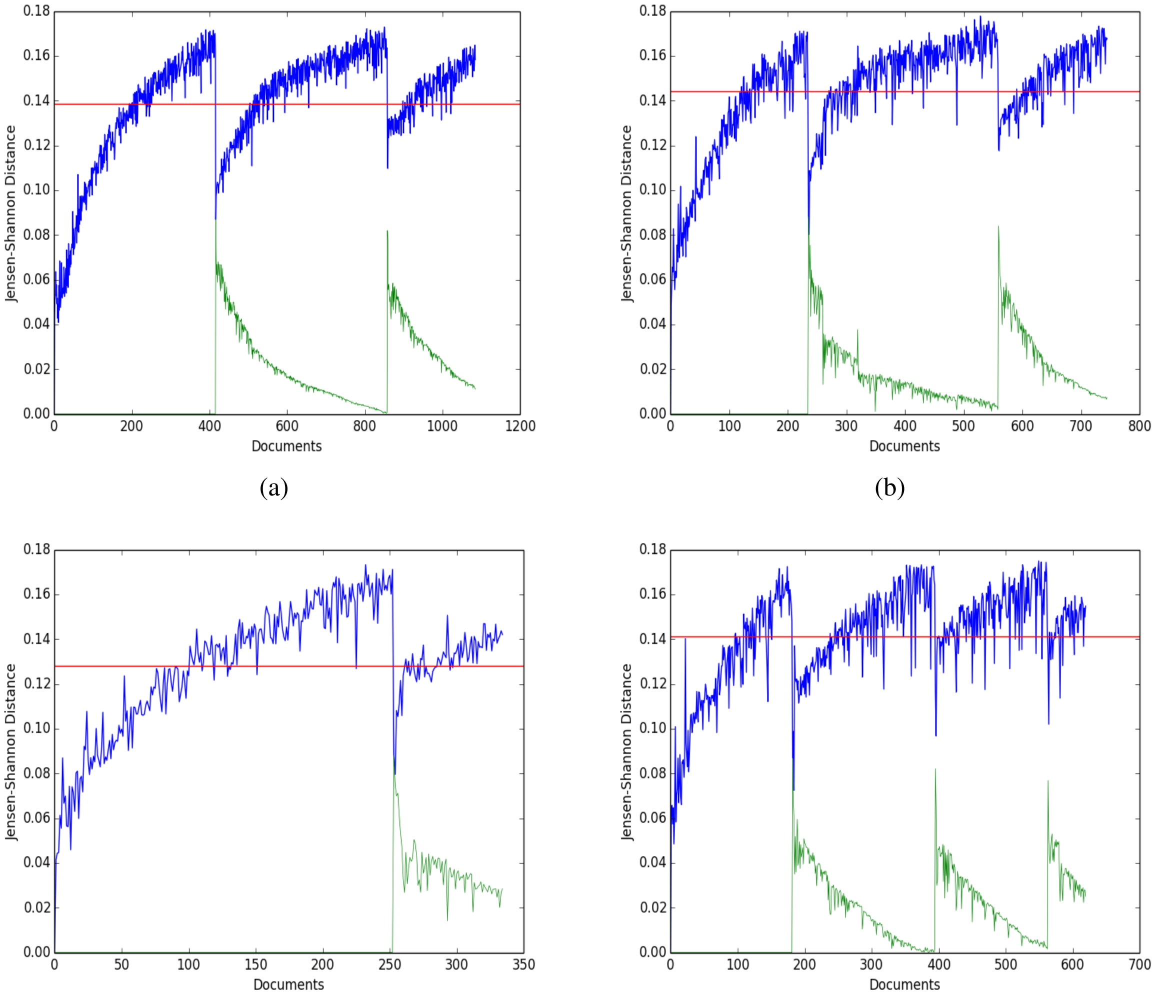
Figure 5.5: The mean value of the Jensen-Shannon distance, computed for each document during training on the Reuters-R8 dataset, for classes (a) *earn*, (b) *acq*, (c) *trade*, (d) *ship*, (e) *grain* and (f) *crude*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



(a)

(b)

Figure 5.6: The mean value of the Jensen-Shannon distance, computed for each document during training on the Reuters-R8 dataset, for classes (a) *interest* and (b) *money-fx*. In all images, the horizontal line is the overall mean value for the distance, computed after training.



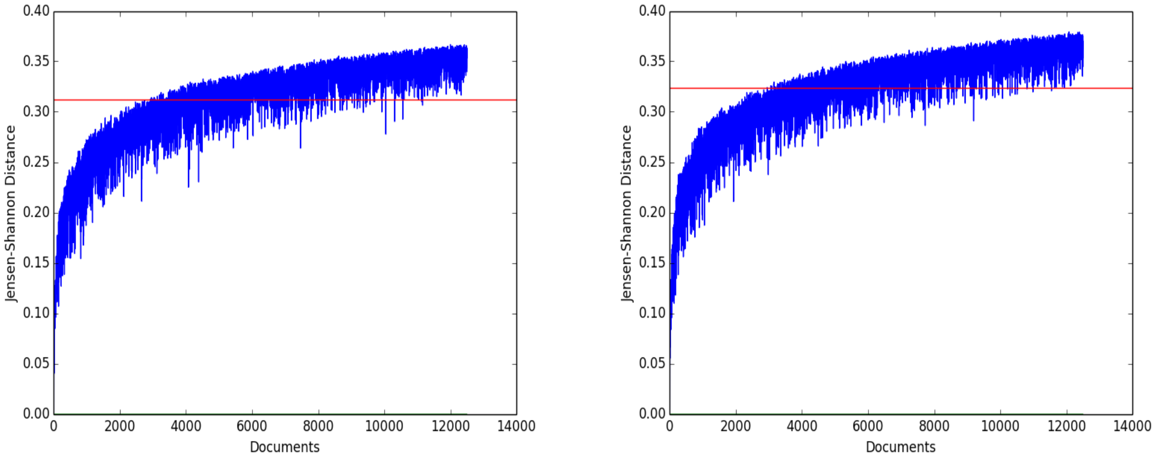
(a)

(b)

(c)

(d)

Figure 5.7: The mean value of the Jensen-Shannon distance, computed for each document during training for all classes of the WebKB dataset: (a) is the performance on class *student*, (b) shows class *faculty*, (c) is class *project* and (d) indicates class *course*. In all images, the horizontal line is the overall mean value for the distance, computed after training; and the thinner line at the bottom indicates the mean standard deviation of the distance, computed among all clusters for each document.



(a) (b)

Figure 5.8: The mean value of the Jensen-Shannon distance, computed for each document during training for all classes of the IMDB dataset: (a) is the performance on class *positive* and (b) indicates the behaviour of DV-INBC on class *negative*. In all images, the horizontal line is the overall mean value for the distance, computed after training.

6 CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

This thesis presented a probabilistic and incremental method to categorize streams of documents, called DV-INBC. The main characteristics of DV-INBC are:

It only needs a single pass over the training data to complete its training For a given set of parameters, the best DV-INBC model is built without needing to iterate repeatedly over the training dataset.

Little knowledge of the data stream is needed to do training Only the number of classes on the dataset is needed. Neither the vocabulary to represent all documents is needed, because DV-INBC builds it as the documents are read from the dataset.

To assess the performance of DV-INBC on document categorization tasks, a set of experiments were conducted involving two tasks: topic classification and sentiment analysis. It was observed that, when compared to the state of the art, DV-INBC had a competitive performance in the topic classification task, when tested against the 20 Newsgroups dataset. However, its performance was worse on the IMDB dataset, which was used to evaluate its performance on the sentiment analysis task. The performance is compatible to what is expected for this task, since the word order is lost when using the bag-of-words model, a piece of information which is very important for that task.

The experiments have also shown that the δ parameter is not robust enough and therefore a selection of the most frequent words to represent a cluster is not the best alternative. This means that other methods to make the comparison between clusters and documents should be considered. Besides, the τ parameter has an indirect effect on DV-INBC, which makes it difficult to understand the real impact on the performance when selecting its value. It indirectly controls the amount of clusters that are created during training, by setting the maximum admissible distance between clusters and documents but the real impact of its value is not clear a priori.

Also, in the other datasets of the topic classification task, it could be seen that the performance of DV-INBC was, at most, equivalent to a conventional Naive Bayes classifier. This was even more visible when testing the model with the Reuters-R8 dataset, where the performance of DV-INBC was much worse than a Naive Bayes model. However, it should be noted that DV-INBC does not use information about the amount of documents of each class, since it was devised as a model to scan over the training data only once, and therefore it is not known a priori how many documents are available on each class. This knowledge is represented by a Naive Bayes classifier as the prior probabilty of each class. As the experiments have presented, if this information is used by DV-INBC, its performance can greatly increase in situations where the class distribution is very skewed, as found in the Reuters-R8 dataset.

Finally, the experiments have shown that the two main characteristics of DV-INBC allow it to be used in situations where the size of the training data is large (since it does not need to iterate over it many times) or when the vocabulary of the dataset is not known a priori. Besides, its main parameters can be estimated by using a subset of the entire training data, which is a fast operation. DV-INBC had a competitive performance to the state of the art on the 20 Newsgroups dataset; a good performance on the Webkb dataset, with results similar to an SVM, and performed poorly in the Reuters-R8 dataset; and a performance worse than the state of the art on the IMDB dataset (which is actually an expected behavior), but still had an error rate of almost half of a more complex approach, which used LDA features to train an SVM.

6.2 Future Work

More work is needed to allow a clearer notion of the real impact of each configuration parameter of the model before training. It is also necessary to find another feature selection method, preferably one capable of identifying occurrence correlations between words inside a cluster (a situation analogous to finding the principal components of a set of n-dimensional points), instead of simply choosing the most frequent words, which was concluded that is not a robust method. Besides, the study of a way of removing the words from the vocabulary of a cluster that do not contribute to a good identification of the topic of the document or to the topic of the cluster itself can help improve DV-INBC.

In order to find a better feature selection method, data mining techniques could be used, such as FP-Growth (HAN; PEI; YIN,) or other similar method. Topic model techniques adapted to work on data streams settings could also be studied and adapted to work with DV-INBC. However, it is important to analyse the execution time of the chosen approach, since the process of selecting features is performed many times during the runtime of DV-INBC. It is also necessary to develop an approach that can handle a large amount of data and make a model of it that can still fit in a reasonable amount of main memory.

To better evaluate the impact of the quality of the data, a pre-processing step could be attached to DV-INBC, prior to the use of a document to train a model. In this step, processes like stemming and stopword removal could be applied. However, it should be noted that such procedures should be optimized to not impose a heavy impact on the computing time. This additional step could also help understand the real performance of DV-INBC by using only data with reduced noise for training.

REFERENCES

AGGARWAL, C. C.; ZHAI, C. **Mining text data**. [S.l.]: Springer, 2012.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, USA: Springer-Verlag New York, Inc., 2006.

BLEI, D. M. Probabilistic topic models. **Commun. ACM**, ACM, New York, USA, v. 55, n. 4, p. 77–84, abr. 2012. ISSN 0001-0782.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435.

CARDOSO-CACHOPO, A. **Improving Methods for Single-label Text Categorization**. 2007. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

DAUPHIN, Y.; BENGIO, Y. Stochastic ratio matching of rbms for sparse high-dimensional inputs. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.], 2013. p. 1340–1348.

DEERWESTER, S. et al. Indexing by latent semantic analysis. **JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE**, v. 41, n. 6, p. 391–407, 1990.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B**, v. 39, n. 1, p. 1–38, 1977.

ENGEL, P. **INBC: An incremental algorithm for dataflow segmentation based on a probabilistic approach**. [S.l.], 2009.

FUGLEDE, B.; TOPSOE, F. Jensen-shannon divergence and hilbert space embedding. In: INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY, 2004. **Proceedings...** [S.l.]. p. 31–.

HAN, J.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2000, Dallas, USA. **Proceedings...** New York, USA: ACM. p. 1–12.

HINTON, G. E. Training products of experts by minimizing contrastive divergence. **Neural Comput.**, MIT Press, Cambridge, USA, v. 14, n. 8, p. 1771–1800, aug. 2002. ISSN 0899-7667.

HOFFMAN, M.; BACH, F. R.; BLEI, D. M. Online learning for latent dirichlet allocation. In: NIPS, 2010. **Proceedings...** [S.l.]. p. 856–864.

HOFMANN, T. Probabilistic latent semantic analysis. In: UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 1999. **Proceedings...** [S.l.]. p. 289–296.

JOHNSON, R.; ZHANG, T. Effective use of word order for text categorization with convolutional neural networks. **CoRR**, abs/1412.1058, 2014.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **The Annals of Mathematical Statistics**, JSTOR, v. 22, n. 1, p. 79–86, 1951.

LAROCHELLE, H. et al. Learning algorithms for the classification restricted boltzmann machine. **The Journal of Machine Learning Research**, JMLR. org, v. 13, p. 643–669, 2012.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. **CoRR**, abs/1405.4053, 2014.

LECUN, Y. et al. Gradient-based learning applied to document recognition. In: **Proceedings of the IEEE**. [S.l.: s.n.]. p. 2278–2324.

MAAS, A. L. et al. Learning word vectors for sentiment analysis. In: THE 49TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES. **Proceedings...** Portland, USA: Association for Computational Linguistics, 2011. p. 142–150.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York, USA: McGraw-Hill, Inc., 1997.

SMOLENSKY, P. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. (Ed.). Cambridge, USA: MIT Press, 1986. chp. Information Processing in Dynamical Systems: Foundations of Harmony Theory, p. 194–281.

SRIVASTAVA, N.; SALAKHUTDINOV, R. R.; HINTON, G. E. Modeling documents with deep boltzmann machines. **arXiv preprint arXiv:1309.6865**, 2013.

WANG, C.; PAISLEY, J. W.; BLEI, D. M. Online variational inference for the hierarchical dirichlet process. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 2011. **Proceedings...** [S.l.]. p. 752–760.

WANG, S.; MANNING, C. D. Baselines and bigrams: Simple, good sentiment and topic classification. In: THE 50TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: SHORT PAPERS - VOLUME 2, Jeju Island, Korea. **Proceedings...** Stroudsburg, USA: Association for Computational Linguistics, 2012. (ACL '12), p. 90–94.

APPENDIX A RESUMO EM PORTUGUÊS

A.1 Introdução

O campo conhecido como *Data Mining* obteve recentes avanços ultimamente, devido principalmente à disponibilidade de diferentes tipos de dados, o que é particularmente verdadeiro para o caso de texto, onde tanto a Web como as redes sociais permitiram a rápida criação de grandes repositórios de dados. A crescente quantidade de dados textuais disponível a partir de diferentes aplicações criou uma necessidade por avanços em design de algoritmos que consigam aprender padrões interessantes a partir destes dados de uma forma dinâmica e escalável (AGGARWAL; ZHAI, 2012). Conforme nosso conhecimento coletivo continua a ser digitalizado e armazenado na forma de mídias como notícias, blogs, páginas da Web, artigos científicos, livros e muitas outras, ferramentas computacionais para organizar, buscar e entender tais quantidades vastas de dados tornam-se necessárias (BLEI, 2012).

No caso de dados textuais, seja minerando um fluxo de dados ou um conjunto fixo de documentos, a tarefa de classificação e retorno de documentos precisa de representações úteis para as informações contidas em tais itens. O fato de que estas informações estão originalmente disponíveis em uma forma não estruturada motiva a pesquisa e o design de algoritmos para resolver estes problemas (SRIVASTAVA; SALAKHUTDINOV; HINTON, 2013). Da mesma forma, aplicações Web (por exemplo, redes sociais) podem criar um fluxo contínuo de grandes volumes de texto, devido à criação simultânea de texto a partir de uma grande variedade de usuários. Tais dados são mais desafiadores para o processo de mineração, já que precisam ser processados no contexto de uma limitação *one-pass*, o que significa que pode ser difícil armazená-los offline para retirar informações úteis dos mesmos e que a tarefa de mineração deve ser realizada conforme os dados chegam (AGGARWAL; ZHAI, 2012).

Uma categoria de algoritmos para descobrir os principais temas que permeiam uma coleção qualquer de documentos é chamada de *Topic Models*. Tais algoritmos podem organizar a coleção de documentos de acordo com os temas descobertos. Da mesma forma, eles podem ser aplicados sobre coleções com grandes quantidades de dados e também para fluxos de documentos (BLEI, 2012). A área de *Topic Modeling* integra algoritmos de *soft clustering* com redução de dimensionalidade. Documentos são associados a um número de tópicos latentes, os quais correspondem tanto a *clusters* de documentos como a representações compactas identificadas na coleção. Cada documento é atribuído aos tópicos com diferentes pesos, o que especifica o grau de pertinência em cada *cluster*. A representação original dos atributos tem um papel chave na definição dos tópicos e na identificação de quais tópicos estão presentes em cada documento. O resultado é uma representação compreensível de documentos que é útil para analisar quais temas estão presentes nos mesmos (AGGARWAL; ZHAI, 2012).

A.2 Motivação

Motivado pela necessidade de algoritmos capazes de processar fluxos de documentos e/ou coleções dos mesmos, este trabalho apresenta uma abordagem para categorizar documentos que pode ser utilizada em tais cenários: DV-INBC, que significa Dynamic Vocabulary Incremental Naive Bayes Clustering. O método necessita apenas de informação sobre a quantidade de classes na coleção de documentos, construindo o vocabulário de forma dinâmica conforme os dados chegam. A representação de cada classe no modelo é uma mistura de distribuições multinomiais, seguindo uma abordagem similar a muitas outras técnicas de *Topic Model*. O DV-INBC é um algoritmo incremental, online e probabilístico que estende o modelo INBC apresentado em (ENGEL, 2009). Essas características fazem de DV-INBC um modelo adequado tanto para categorizar fluxos de documentos como para processar coleções de textos como se fossem fluxos, o que pode ser observado pelos experimentos apresentados neste trabalho. Além disso, o algoritmo apresentado pode ser útil em cenários em que o tamanho dos dados é muito grande para sofrer repetidas iterações como outras técnicas fazem, bem como em situações onde há um fluxo contínuo de documentos para ser processado.

Portanto, a principal contribuição deste trabalho é o desenvolvimento de um novo algoritmo para classificação de documentos, chamado DV-INBC, e a sua avaliação em *datasets* populares para a tarefa. Os resultados observados nos experimentos mostram que o modelo é promissor, embora ainda existam melhoramentos a serem feitos.

A.3 Conclusões

Este trabalho apresentou um método probabilístico e incremental para categorização de documentos, chamado DV-INBC. As principais características do modelo são:

Apenas uma única época sobre os dados de treinamento é necessária para completar o processo de treinamento Para um dado conjunto de parâmetros, o melhor modelo DV-INBC é construído sem a necessidade de iterar repetidamente pelo *dataset* de treinamento.

São necessárias poucas informações sobre os dados de treinamento Só é necessário saber o número de classes no *dataset*. Nem o vocabulário para representar os documentos é preciso, pois DV-INBC o constrói conforme os documentos são lidos.

Para avaliar a performance do DV-INBC em tarefas de categorização de documentos, experimentos foram realizados envolvendo duas tarefas: classificação de tópicos e análise de sentimento. Foi observado que, quando comparado com o estado da arte, DV-INBC teve performance competitiva na tarefa de classificação de tópicos, quando foi avaliada sua performance no *dataset* 20 Newsgroups. Entretanto, sua performance foi pior quando avaliada sobre o *dataset* IMDB, o qual foi utilizado para analisar seu desempenho na tarefa de análise de sentimento. Nesta tarefa, sua performance é compatível com o que se espera de um modelo baseado em *bag-of-words*, já que a ordem das palavras é uma informação muito importante em análise de sentimentos.

Os experimentos também mostraram que o parâmetro δ não é robusto o suficiente e que portanto a seleção das palavras mais frequentes para representar um *cluster* não é a melhor alternativa. Isto significa que outros métodos para realizar a comparação entre *clusters* e documentos deve ser considerada. Além disto, o parâmetro τ tem uma influência indireta sobre DV-INBC, o que torna difícil entender o real impacto na performance do modelo ao selecionar seu valor. Este parâmetro controla indiretamente a quantidade de *clusters* que é criada durante o treinamento, através do controle da máxima distância admitida entre *clusters* e documentos, entretanto o real impacto de seu valor não é claro *a priori*.

Da mesma forma, nos outros *datasets* utilizados para a tarefa de classificação de tópicos, foi possível observar que a performance do DV-INBC foi, no máximo, equivalente a um classificador Naive Bayes tradicional. Isto foi mais visível ainda ao testá-lo no *dataset* Reuters-R8, onde sua performance foi muito pior que tal classificador. Entretanto, deve ser notado que DV-INBC não usa informação sobre a quantidade de documentos em cada classe, já que foi projetado para ler o conjunto de treinamento apenas uma vez, e portanto tal número não é conhecido *a priori*. Esta informação é representada em um classificador Naive Bayes como a probabilidade *a priori* de cada classe. De acordo com os experimentos, foi observado que se esta informação é utilizada, a performance do DV-INBC pode melhorar consideravelmente, principalmente em situações em que a distribuição de documentos por classe é bem enviesada, tal como no *dataset* Reuters-R8.

Finalmente, os experimentos mostraram que as duas principas características do DV-INBC o permitem ser usado em situações onde o tamanho do conjunto de treinamento é grande (já que não é necessário iterar sobre o mesmo muitas vezes) ou quando não se sabe *a priori* o vocabulário do dataset. Além disto, seus principais parâmetros podem ser estimados usando um subconjunto do *dataset* original, o que é uma operação rápida. DV-INBC tem uma performance competitiva com o estado da arte no *dataset* 20 Newsgroups; uma boa performance no *dataset* WebKB, com resultados similares a uma SVM e um desempenho ruim no *dataset* Reuters-R8; uma performance pior que o estado da arte foi obtida no *dataset* IMDB (um comportamento esperado), ainda assim sua taxa de erro foi quase a metade de outra abordagem mais complexa, que utilizou atributos obtidos a partir de um modelo LDA para treinar uma SVM.

A.4 Trabalhos futuros

Mais trabalhos são necessários para se atingir uma noção mais clara do real impacto de cada parâmetro de configuração do modelo. Também é necessário encontrar outro método de seleção de atributos, preferencialmente algum capaz de identificar a correlação das ocorrências entre palavras dentro de um *cluster* (situação análoga a encontrar os componentes principais de um conjunto de pontos com n dimensões), em vez de simplesmente selecionar as palavras mais frequentes, o que foi concluído como sendo um método não robusto. Além disto, o estudo de uma forma para remover palavras do vocabulário que não contribuem para uma boa identificação tanto do tópico do documento como do *cluster* podem ajudar a melhorar o desempenho do DV-INBC.

Para encontrar um método de seleção de atributos melhor, técnicas de *Data Mining* podem ser utilizadas, tais como FP-Growth (HAN; PEI; YIN,) ou métodos similares. Técnicas de *Topic Model* adaptadas para funcionar em cenários de fluxos de dados podem ser estudadas e adaptadas também para funcionar no DV-INBC. Entretanto, é importante analisar o tempo de execução da abordagem escolhida, já que o processo de seleção de atributos é realizado muitas vezes durante a execução do DV-INBC. Da mesma forma é preciso desenvolver uma forma de lidar com o grande volume de dados e criar um modelo dos mesmos que possa ajustar-se em uma quantidade razoável de memória principal.

Para melhor avaliar o impacto da qualidade dos dados, um passo de pré-processamento poderia ser adicionado ao DV-INBC, previamente ao processamento de um documento durante o treinamento. Neste passo, operações como *stemming* e remoção de *stopwords* poderiam ser aplicadas. No entanto, deve ser notado que tais processos devem ser otimizados para não causarem aumentos significativos no tempo de execução do DV-INBC. Este passo adicional poderia também ajudar a entender qual a performance real do modelo, utilizando-o somente sobre dados com pouco ruído.