Positive Unlabeled Learning: Adapting *NMF* for text classification

Lucas S. S. Nunes[1],**Thiago de P. Faleiros**[1], **Rafael G. Rossi** [2]

[1]Departamento de Ciência da Computação – Universidade de Brasília (UnB)
CEP 70910-900 – Brasília – DF – Brazil

[2]Ifood

221106132@aluno.unb.br, thiagodepaulo@unb.br, rgr.rossi@gmail.com

Abstract. *Due to the overwhelming data generation that surpasses human evaluation capacity, manually labeling data for training machine learning models is becoming increasingly impractical. This article focuses on analyzing techniques to address the challenges of Positive Unlabeled Learning (PUL). To this end, we propose structural adaptations to the Non-Negative Matrix Factorization (NMF) algorithm, specifically tailored for PU data (NMFPUL). We compare NMFPUL with state-of-the-art techniques to identify improvements in the performance of textual data classification. Our study reveals that NMFPUL consistently outperforms most baseline algorithms across diverse document collections even with a limited number of labeled documents, and mainly on these situations.*

1. Introduction

The incorporation of new technologies into the daily lives of individuals and institutions has led to an exponential increase in the volume of data generated [Naeem et al. 2021]. This fact was intensified in the 2020/2021 biennium due to the COVID-19 pandemic, as a large portion of everyday human activities that were once carried out in person were replaced by virtual interactions. The most common way of storing data is through the **textual** format, such as in magazines, articles, web pages, social media and application *logs*, product and service evaluations (*reviews*), among others, and the search for patterns on this data is called text mining. While data mining deals with structured data that are generated in software applications, spreadsheets, structured databases, text mining must deal with unstructured data, that is, data in textual format as in the examples mentioned at the beginning of this paragraph [Li et al. 2022].

Text mining algorithms can use supervised, semi-supervised, or unsupervised machine learning [Kowsari et al. 2019]. The most common form of *machine learning* is *supervised learning*, but this technique requires a large proportion of the data to be previously labeled so that the supervisor's responses are guided [Mahesh 2020]. Furthermore, having a previously labeled and a large enough dataset to train the model properly is usually not straightforward. Thus, semi-supervised learning allows finding solutions to problems with less prior information. In semi-supervised learning, labeled data are combined with unlabeled data to perform the learning [van Engelen and Hoos 2020].

One of the most studied problems in machine learning is binary classification, which, with an almost entirely labeled dataset (or entirely), a model should be trained to learn to classify data within the positive or negative classes. ***PU Learning* (*Positive Unlabeled Learning – PUL*)** is a variant of this problem, and one of its main differences from the binary classification problem is that it assumes the use of unlabeled data on the training set [Bekker and Davis 2020]. In addition to these, some techniques associated with *PU Learning* are close to semi-supervised machine learning [Faleiros et al. 2020]. One of the reasons *PU learning* has been extensively studied recently is that *PU* data appear in various important applications in fields such as medicine, nanotechnology, fake news detection, digital advertising, and scientific documents.

The problems that surge when applications generate massive and unstructured text data are still to have homogeneous and efficient solutions. Typically, text classification is performed by using supervised multi-class learning techniques. However, labeled text must be provided to perform a supervised approach technique. Therefore, there are a some weaknesses on this approach that need to be addressed, in order to achieve better text classification, such as: i) the neglect of unlabeled examples to perform a learning process; ii) the high cost of labeling data; iii) the difficulty of applying a machine learning to sparse data, specially when using textual data.

Efficiently classifying positive and unlabeled data, which often exhibit numerous features and complex relational interdependencies, is challenging. However, addressing data sparsity and leveraging the available information can lead to promising outcomes. This study aims to apply Non-negative Matrix Factorization (*NMF*) to classify positive and unlabeled (PU) data [Jaskie and Spanias 2019]. Given the high cost associated with labeling data, it is imperative to employ efficient methods for classifying textual data sources that contain a limited amount of labeled data and a substantial amount of unlabeled data [Li et al. 2016]. To illustrate the distinction between PU, semi-supervised, and supervised learning, refer to Figure 1.

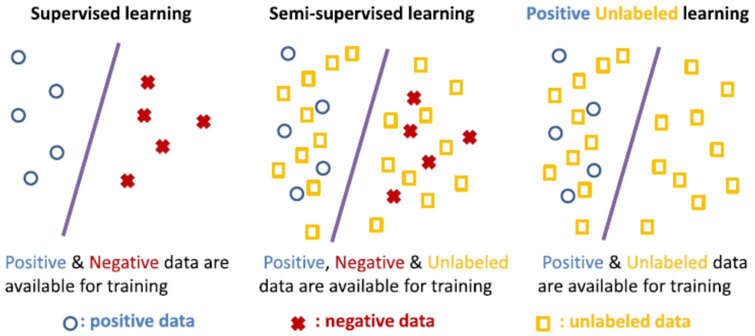
Supervised learning
Semi-supervised learning
Positive Unlabeled learning
Positive & Negative data are available for training
Positive, Negative & Unlabeled data are available for training
Positive & Unlabeled data are available for training
○: positive data
✖ : negative data
□ : unlabeled data

Figure 1. Exemple of supervised learning, semi-supervised learning and positive unlabeled learning. Source: Figure extracted from (Wu et al., 2021).

This paper introduces *NMFPUL* (Non-negative Matrix Factorization for Positive Unlabeled Learning), an algorithm adapted from Non-negative Matrix Factorization (*NMF*) for text classification in the context of Positive Unlabeled (*PU*) data. *NMFPUL* addresses the challenge of limited labeled data, a common issue in various real-world applications. By leveraging the capabilities of NMF, a well-established technique for dimensionality reduction and topic extraction from text datasets, we propose a novel algorithm for text classification that provides a fresh perspective on the potential of matrix factorization in supporting classification tasks with reasonable performance.

The paper is organized as follows. In Section 2, we provide a comprehensive overview of the basic concepts necessary for understanding this work. Section 3 introduces the *NMFPUL* algorithm, presenting its key components and methodology. Section 4 outlines the research methodology employed to conduct our experiments. Section 5 presents the results of our investigation, highlighting the impact of different Positive Unlabeled Learning (*PUL*) algorithms. Finally, in Section 6, we summarize the findings of our study and discuss potential avenues for future research.

2. Foundations and related work

In this section, we present the foundations necessary to support the methodology used in this article. Notations, definitions, and background are described in this section, and related work regarding studying positive unlabeled learning algorithms, focusing on aplying textual data.

2.1. Notations and definitions

A *PU* dataset is represented as a set of triples, a specialization of a 3-tuple of elements, (x, y, l), where x is an attribute vector, y is the variable that indicates the class, and l, a binary variable indicating whether the example is labeled. Table 1 contains the notations that will be used throughout this section.

Table 1. Mathematical notations used in this chapter.

Notation	Description
(x, y, l)	PU dataset triple of elements
x	Attribute vector
y	Variable indicating the class of the example
l	Variable indicating the labeling of the example
X	Attribute vector set
α	Class prior, where $P(y = 1)$
$e(x)$	Propensity function, $P(s = 1\|y = 1, x)$
$f(x)$	Probability density function
$f_+(x)$	Probability density function of the positives examples
$f_-(x)$	Probability density function of the negatives examples
$f_l(x)$	Probability density function of the labeled examples
$f_u(x)$	Probability density function of the unlabeled examples

For a given example, if it is positive or belongs to the positive class, then $y = 1$, and if it is negative or belongs to the negative class, $y = 0$. In a PU dataset, if the data is labeled, we have that $l = 1$; if it is not labeled, $l = 0$. We also call the main class the positive class, that is, the target class of the problem, here defined as $\alpha = P(y = 1)$, where P indicates the probability function [Bekker and Davis 2020].

In a *PUL* problem, if an example is labeled, it is known to belong to the positive class. This implies that we can confidently state that $P(y = 1|s = 1) = 1$. However, when an example is not labeled, it is uncertain whether it belongs to the positive class, the main class of interest, or the negative class. In other words, unlabeled examples can potentially belong to either the positive or negative class.

2.2. Text data representation

The textual representation indicates the way data will be formatted and displayed. Usually, the text datasets are composed of documents containing unstructured format. Therefore, to apply certain machine learning algorithms, performing a transformation to represent textual data in numerical vectors is necessary.

The ***Bag of Words - BoW*** is a traditional technique that transforms texts into a fixed-length vector, where each vector entry indicates the occurrence or absence of a word from the text's vocabulary. For each document in a collection, this vector can be created. In this sense, the repetition of words is not considered. The vector will comprise the words from the vocabulary, not the corpus, where the latter term indicates the totality of words in the texts. The order of occurrence of words is also not considered.

Another useful technique to apply to text data to represent it as a vector is **TF-IDF (Term Frequency and Inverse Document Frequency)**. *TF-IDF* can be defined as calculating how much a word is relevant to a set of documents or corpus. Term Frequency (*TF*) refers to the frequency of the word t appearing in a document d. It quantifies the number of occurrences of the word within the document. The inverse document frequency (*IDF*) measures a word's informativeness by determining its commonness or rarity across all documents. It provides insight into how much valuable information a word contributes based on its prevalence or scarcity in the entire document collection. Therefore, *TF-IDF* is the product of *TF* and *IDF*, resulting in:

$$\text{TF-IDF}(t, d) = TF(t, d) \cdot IDF(t) \tag{1}$$

The product of term frequency (*TF*) and inverse document frequency (*IDF*) determines the weight assigned to a term t. This weighting mechanism gives higher importance to terms with a high frequency within a specific document and a low frequency across the entire collection of documents. Consequently, the implementation yields a matrix of documents and terms, where each cell represents the *TF-IDF* value of the related term within the document collection.

2.3. Positive Unlabeled Learning

PUL algorithms perform learning by using a set of labeled positive documents and unlabeled documents to train a classifier (inductive semi-supervised learning) or they classify the unlabeled documents (transductive semi-supervised learning) [Carnevali et al. 2021].

In semi-supervised learning, unlabeled data is used in the training process when it exists, but usually, some labeled data from all classes are available [Bekker and Davis 2020]. On the other hand, PU Learning uses unlabeled data in the learning process, and generally, the labeled data belongs to only one class. Unlike other learning techniques, PU Learning uses only a small portion of labeled positive data, and none of the negative data is labeled. Then, the classifier's training must be performed from positive and unlabeled data.

Several conditions must be considered when conducting a study and utilizing Positive and Unlabeled Learning (PUL) methods to classify data. These conditions are crucial for applying PUL to a problem and simulating a real machine learning scenario using positive and unlabeled data.

Firstly, it is essential to acknowledge that all examples in the dataset that are not labeled are considered to belong to the negative class. In other words, any unlabeled example is assumed to be part of the negative class rather than the main or target class. This assumption is a fundamental premise for the training process and the labeling mechanism, ensuring consistency in the interpretation and handling of unlabeled instances [Wang et al. 2022]. As a consequence of the first condition, any labeled data belongs to the positive class. Therefore, for the model training process, the labeled examples from the positive class will be important to identify the most likely negative examples, the Reliable Negatives.

Another assumption is that there should be a clear way to separate the classes. A parameter or set of parameters should perfectly identify the difference between the classes. The fourth assumption we should make is that examples close to each other are more likely to have the same label, a condition fundamental for the *PUL* method called the Two-step Technique. This smoothness property is commonly used in graph approaches [Carnevali et al. 2021, Wu et al. 2021].

Some approaches that use the Two-step techniques, which are the most frequently applied, are Unbiased Positive-Unlabeled (*UPU*) [Yang et al. 2020], Spy Expectation-Maximization (*Spy-EM*) [He et al. 2020], Non-negative Positive-Unlabeled (*nnPU*) [Ji et al. 2023], Positive and Unlabeled Learning by Label Propagation (*PU-LP*) [Jaemin et al. 2022, Ma and Zhang 2017], Rocchio Support Vector Machine(*RC-SVM*) [Li and L 2003], Label Propagation for Positive and Unlabeled Learning (*LP-PUL*) [Carnevali et al. 2021]. The last three algorithms referenced will be used as baselines for this study.

The ***RCSVM*** algorithm is based on Support Vector Machines (*SVM*) and leverages the Rocchio method [Li and L 2003] to generate prototypes for both positive and unlabeled data. By comparing the similarity between documents and positive versus unlabeled data, *RCSVM* identifies reliable negative documents that are more similar to the unlabeled data. The *RCSVM* algorithm has a time complexity of $O(n^3)$.

On the other hand, ***PU-LP***, which has a time complexity of $O(n^2 logn)$, [Jaemin et al. 2022] is a graph-based algorithm that utilizes a similarity matrix created using the *k-Nearest Neighbors* approach. It identifies the unlabeled nodes with the lowest similarity scores and designates them as reliable negative examples. Conversely, nodes with higher similarity scores are assigned to the positive set, which is then used to label the remaining documents, transforming the problem into a positive-negative classification task.

Lastly, ***LP-PUL***, also employing a graph-based approach, and having the same time complexity as *PU-LP*, involves three steps: 1) constructing a document graph based on similarity measures, 2) inferring reliable negative documents based on the neighborhood of nodes in the graph, and 3) applying a label propagation mechanism using the positive and negative documents to classify the remaining unlabeled documents [Carnevali et al. 2021]. These algorithms offer distinct methodologies for handling positive and unlabeled data, each with advantages and considerations.

We also used a One-Class Learning technique for algorithm performance comparison, which uses all the positive documents to build a classifier. In contrast, all the other documents belong to the negative class [P. Tan and Kumar 2019]. In this work, we used a **K-Means** based algorithm, having a time complexity of $O(n^2)$, where the positive documents are divided into groups, and each group has a centroid, which is calculated based on the average of document vectors from the group. The similarity of each new document is compared to the centroids to define the class that document will belong to.

2.4. Labeling Mechanism

To understand how the labeled positive examples are selected, we must understand how these examples originate from the original dataset. There are two ways to identify this issue: the data come from just one dataset, which is an independently and identically distributed (i.i.d.) sample from the real population, known as the single set scenario; or the data originate from two datasets, one of them consisting only of positive examples from the population and the other dataset, made up of only unlabeled examples, is an independently and identically distributed (i.i.d.) sample from the real population, known as the case-control scenario [Bekker and Davis 2020].

For the purpose of studying the labeling mechanism, focus will be given to the first defined scenario, where a fraction of the positive examples are labeled, following the labeling propensity function, as in equation below:

$$X \approx f(x) \tag{2}$$

$$X \approx \alpha f_+(x) + (1-\alpha) f_-(x) \tag{3}$$

$$X \approx \alpha e(x) f_l(x) + (1 - \alpha e(x)) f_u(x) \tag{4}$$

where f(x) is the probability density function of the population, and e(x) is the propensity function, which indicates the probability of a positive example being labeled.

Based on this, we can understand how a labeling mechanism can be applied to PU data. We start by defining the probability density function of an example being labeled concerning the probability density of it being positive:

$$f_l(x) = P(x|s=1, y=1) \tag{5}$$

$$f_l(x) = \frac{P(x|s=1, y=1)}{P(s=1, y=1)} P(x, y=1) \tag{6}$$

It is also understood that the example is unlabeled if it is a negative or positive example but was not selected by the labeling mechanism to be labeled. Thus, to allow for direct learning from PU data, one must understand and define the approach used concerning the labeling mechanism and the distribution of examples in the classes. There are two main approaches for the labeling mechanism, *SAR* and *SCAR*.

The labeling mechanism, known as **Selected at Random (*SAR*)**, operates under the hypothesis that selecting positive examples for labeling depends on their attributes. This makes *SAR* the most generalist approach, as it acknowledges the influence of inherent biases in various real-world applications [Bekker and Davis 2018]. For instance, in tasks such as spam detection in emails or product review analysis, the selection of positive examples to label may depend on the compelling nature of the text itself

[Bian et al. 2021, Wu et al. 2020]. Similarly, recommendation systems might be influenced by the order in which the initial products or services are presented, which can bias subsequent recommendations. *SAR* is considered a weakened form of ***SCAR*** (Select and Classify at Random) [Jaskie and Spanias 2019].

The labeling mechanism referred to as **Selected Completely at Random (*SCAR*)** involves selecting a subset of positive examples to label. In this scenario, every positive example has an equal probability (c) of being chosen for labeling. Unlike *SAR, SCAR* operates under the assumption that any bias present in the labeled set will be transferred to the model's bias. Therefore, the random selection of positive examples for labeling aims to minimize selection bias as much as possible [Bekker et al. 2019].

2.5. Non-negative Matrix Factorization

NMF, Non-Negative Matrix Factorization, is an algorithm or set of algorithms where a matrix $V \in \mathbb{R}+^{dxn}$ is factored in a way to produce two matrices: $W \in \mathbb{R}+^{dxk}$ and $H \in \mathbb{R}_{+}^{kxn}$, where the three matrices have non-negative elements, as in:

$$V = WH^T \tag{7}$$

When analyzing textual data, we can represent the data using matrices. In this case, the matrix V has documents as rows and the vocabulary of words as columns. This matrix is constructed using the bag-of-words approach with *TF-IDF* word weighting. In other words, the matrix is created by representing each document as a collection of words, and the *TF-IDF* technique is applied to assign weights to these words based on their frequency within the document and their significance across the entire document collection. The matrix W represents the documents and their topic relationships, indicating how each relates to different topics. The second matrix, H, reveals the contribution of each word to the topics. W and H matrices are commonly used in topic modeling techniques [Lee and Seung 2000].

Non-negative Matrix Factorization (*NMF*) distinguishes itself from other factorization methods, such as Singular Value Decomposition (*SVD*), by imposing non-negativity constraints on the data. *NMF* is primarily employed for feature extraction and dimensionality reduction. While NMF is often compared to Principal Component Analysis (*PCA*) as both methods reduce dimensionality, *NMF* differs because it is specifically designed for non-negative and sparse data.

NMF has found widespread application in medical research, albeit with certain adaptations. However, in situations where interpretability holds significant importance, such as in biomedical research, the unsupervised nature of *NMF* introduces certain limitations. To address this, a masking technique has recently been proposed, which allows for a more interpretable decomposition process within the *NMF* algorithm [X. Lin 2020]. This adaptation enhances the utility of *NMF* in scenarios where interpretability is critical.

Additionally, *NMF* involves the task of finding matrices W and H such that the approximation $V \approx WH$ holds. The quality of this approximation is typically evaluated using an objective function, such as the Euclidean distance. However, in this study, we plan to utilize the **Kullback-Leibler divergence** (KL divergence) [Hien and Gillis 2020] as a measure of the reconstruction error. The KL divergence provides a more appropriate metric for comparing the dissimilarity between probability distributions, which aligns well with the data's nature and the study's goals. The Kullback-Leibler divergence is given by:

$$D(V||WH) = \sum(V \cdot \log\left(\frac{V}{WH}\right) - V + WH) \tag{8}$$

The core of the *NMF* algorithm, when utilizing KL divergence, lies in finding update rules for matrices W and H that minimize the KL divergence at each iteration until convergence. To achieve this, we apply multiplicative update rules to matrices W and H, which can be expressed as follows:

$$W = W \odot \left(\frac{V}{WH + \epsilon} H^T\right) / \left(M_1 H^T\right) \tag{9}$$

$$H = H \odot \left(W^T \frac{V}{WH + \epsilon}\right) / \left(W^T M_1\right) \tag{10}$$

where the symbol $\odot$ denotes element-wise multiplication, M_1 is a matrix with all elements being 1, and ϵ is a small constant to avoid division by zero.

The provided update rules are applied iteratively, beginning with randomly initialized non-negative matrices W and H. The iteration continues until the KL divergence minimization problem converges or the maximum number of iterations is reached. This iterative process aims to refine the matrices W and H to better approximate the original matrix V and minimize the reconstruction error measured by KL divergence.

3. Methodology

Our proposed method, Non-negative Matrix Factorization for Positive Unlabeled Learning (*NMFPUL*), leverages a numerical vector representation of text data, such as bag-of-words and *TF-IDF*, and adapts the concept of *NMF* to classify unlabeled data in a positive and unlabeled (PU) setting. The algorithm, described in Algorithm 1, consists of four main steps:

1. Construction of the **document-term matrix** using the bag-of-words and *TF-IDF* approach.
2. Formation of the **document-topic** and **topic-term matrices** during the initial iteration. The labeled documents are selected using the **Selected Completely at Random (*SCAR*)** mechanism, randomly choosing positive examples for labeling.
3. To ensure that positively labeled documents have the highest value in the first position (first topic) only. This step enhances the separation between positive and unlabeled data. A positive non-zero value ϵ is assigned for all remaining dimensions, representing a very low value. We set this value to 0.001 in our experiments to ensure its small magnitude.
4. Classification of the unlabeled documents using the updated document-topic and topic-term matrices.

By combining the principles of *NMF* with the specific requirements of PU learning, our *NMFPUL* method aims to classify unlabeled data within the PU framework effectively.

Algorithm 1 *NMFPUL* – Non-negative Matrix Factorization for Positive Unlabeled Learning

```
Require: Document Collection, matrix representation V, $number\_labeled\_documents$, $number\_topics$, $max\_iteration$, $tolerance$, number of topics $k$
  function NMFWithUpdate($V$, $labeled\_documents\_indexes$, $k$, $\epsilon = 0.001$)
    Initialize $k$ dimensions of matrices $W$ and $H$ with random values
    $positive\_class\_index \leftarrow 0$
    for $n$ in $range(1, max\_iteration + 1)$ do
      Update $W$ and $H$ according to Multiplicative Update
      $W[labeled, 0] \leftarrow \max(W)$
      $W[labeled, 1:] \leftarrow \epsilon$
      Calculate Kullback-Leibler divergence between $V$ and $W \times H$ as $error$
      if $error < tolerance$ then
        break
      end if
    end for
    return $W, H$
  end function
```

The initial step involves preprocessing the data according to the following sequence: Firstly, the classes and the text data are assigned to respective variables to facilitate the subsequent vectorization process using *TF-IDF*. Next, one of the classes is randomly selected as the positive class, while the remaining classes are designated as negatives. The indices corresponding to the positive class are shuffled from the specified number of initially labeled documents, and the first $number\_labeled\_documents$ indices are chosen as the positive labeled documents. The remaining positive class documents are considered unlabeled.

The algorithm *NMFPUL*, Algorithm 1, receives as input the matrix document-term V and the indexes of the labeled documents. Some values are important to be initialized before instantiating the algorithm: the number of topics that the *NMF* portion algorithm must build the matrices W and H, the number of labeled documents of the positive class, the maximum number of iterations for the convergence, the tolerance which may stop the iterative process if the divergence of *Kullback-Leibler* reaches it before the maximum number of iterations. Also, NMFPUL has a time complexity of the NMF algorithm which is $O(nmk)$, where *n* and *m* are the dimensions of the matrix *V* and *k* is the number of topics defined.

4. Experimental Evaluation

On evaluating the performance of the adapted *NMF* developed on this study, we compare it with other state-of-the-art *PUL* algorithms as well as usual text classification algorithms. We use the benchmarking of evaluation of the study from [Carnevali et al. 2021], while adapting some parameters to better include the analysis of the application of a dimensionality reduction tecnique to positive unlabeled learning problems.

4.1. Datasets

The experiments are applied to nine datasets: text collections composed of a collection of terms and a class for each document [Rossi et al. 2013]. Those collections are related to several areas, such as medical documents, scientific documents, news, and product reviews. The dataset information is summarized at **Table 2**.

Table 2. Document Collection datasets characteristics.

Dataset	Domain	# Docs	# Terms	Avg Terms	# Classes
CSTR	Scientific Reports	299	1726	54.27	4
Fbis	News Articles	2463	2001	159.24	17
Oh0	Medical Documents	1003	3183	52.5	10
Oh15	Medical Documents	3101	54142	17.6	10
Re0	News Articles	1504	2887	51.73	13
Re1	News Articles	1657	3759	52.70	25
SyskillWebert	Web Pages	334	4340	93.16	4
Tr11	TREC Documents	414	6430	281.66	9
WAP	Web Pages	1560	8461	141.33	20

4.2. Experiment configuration and evaluation criteria

The experiments are conducted by considering different parameters and algorithms from the state-of-the-art in Positive and Unlabeled Learning (*PUL*) and One-Class learning techniques, as reported in [Carnevali et al. 2021]. Also, the experiments where developed within using *python* along the packages *Pandas, Numpy and Scikitlearn* and using a local computational environment in order to compile and execute the experiments. To begin, the document-term matrix V is constructed by vectorizing the word sequences of each document and applying the *TF-IDF* transformation.

A simulated Positive Unlabeled Learning (*PUL*) scenario is considered by applying the process to multi-class text collections as we adopted an iterative approach. Since we have a multi-class dataset, a single class from the text collection is designated as the positive class. In contrast, the rest are designated as the negative class. After this, we define a variable to *NMFPUL* (*NMF* for Positive Unlabeled Learning) to select the number of labeled documents in the positive class selected. The values for this parameter are $D_+ = \{1, 5, 10, 20, 30\}$, and the labeled documents are selected randomly, as defined in Section 2.4, through the labeling mechanism *SCAR*. The remaining documents, i.e., those belonging to the negative class and the remaining positive, are left unlabeled.

After randomly selecting the positive class and labeling positive documents, the matrices W and H are initialized, with the number of topics (k) determining the dimensions of these matrices. The suppress function is applied to ensure the labeled documents in matrix W are properly represented. It forces the highest value to be placed in the first position of the corresponding row for the labeled documents.

Since the *NMFPUL* algorithm builds upon the principles of *NMF*, it is essential to specify the maximum number of iterations for convergence. For this experiment, a maximum of 300 iterations is defined. The Kullback-Leibler (KL) divergence is chosen as the objective function to measure the approximation of the matrices [Hien and Gillis 2020].

This divergence metric quantifies the dissimilarity between two probability distributions and is well-suited for evaluating the performance of *NMFPUL*.

The random selection of documents for the positive set can influence the classification outcome; ten trials are conducted, each time selecting different labeled documents. We average the results of these trials to mitigate the effect of randomness. Also, for the steadiness of the experiments, this whole configuration is executed three times for each dataset by changing the parameter of the F1-score for *NMF* when it is applied to multiclass problems. The instances $(micro, macro, weighted)$ are used to calculate F1.

5. Results

To evaluate the performance of *NMFPUL*, we compare the F1-score against other algorithms using the same datasets. We utilize the results presented in [Carnevali et al. 2021] for the selected document collections to analyze the classification performance. Specifically, we experiment with each average parameter of the F1-score.

To ensure a fair comparison with baseline algorithms, we calculate the average the three results obtained for each dataset and the number of labeled documents. The comprehensive results can be observed in Tables 3 to 11. These tables provide insights into the comparative performance of *NMFPUL* and other algorithms across different datasets and numbers of labeled documents.

Table 3. F1 Score values for different algorithms on Document Collection CSTR.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.49	0.61	0.68	0.65	0.51
PU-LP	0.55	**0.74**	**0.78**	0.7	0.69
RCSVM	0.02	0.12	0.29	0.52	0.39
LP-PUL	0.61	0.69	0.77	**0.79**	**0.8**
NMFPUL	**0.632**	0.679	0.715	0.735	0.741

Table 4. F1 Score values for different algorithms on Document Collection Oh0.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.42	0.61	0.68	0.71	0.69
PU-LP	0.28	0.54	0.61	0.6	0.59
RCSVM	0.01	0.19	0.37	0.5	0.59
LP-PUL	0.51	0.67	0.7	0.73	0.71
NMFPUL	**0.707**	**0.726**	**0.744**	**0.761**	**0.771**

Table 5. F1 Score values for different algorithms on Document Collection Oh15.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.38	0.53	0.60	0.63	0.65
PU-LP	0.25	0.51	0.52	0.57	0.51
RCSVM	0.01	0.10	0.26	0.43	0.53
LP-PUL	0.41	0.56	0.60	0.63	0.65
NMFPUL	**0.717**	**0.729**	**0.725**	**0.726**	**0.729**

Table 6. F1 Score values for different algorithms on Document Collection Fbis.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.40	0.50	0.53	0.52	0.51
PU-LP	0.24	0.44	0.48	0.42	0.39
RCSVM	0.12	0.43	0.51	0.56	0.58
LP-PUL	0.41	0.51	0.53	0.52	0.5
NMFPUL	**0.740**	**0.740**	**0.741**	**0.745**	**0.746**

Table 7. F1 Score values for different algorithms on Document Collection Re0.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.40	0.53	0.59	0.39	0.41
PU-LP	0.28	0.45	0.45	0.39	0.36
RCSVM	0.12	0.37	0.52	0.39	0.41
LP-PUL	0.32	0.41	0.45	0.50	0.45
NMFPUL	**0.714**	**0.722**	**0.730**	**0.735**	**0.740**

Table 8. F1 Score values for different algorithms on Document Collection Re1.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.41	0.59	0.60	0.41	0.37
PU-LP	0.27	0.45	0.50	0.43	0.38
RCSVM	0.09	0.23	0.36	0.29	0.31
LP-PUL	0.39	0.54	0.60	0.70	0.62
NMFPUL	**0.755**	**0.757**	**0.762**	**0.767**	**0.768**

Table 9. F1 Score values for different algorithms on Document Collection Tr11.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.50	0.57	0.55	0.41	0.30
PU-LP	0.40	0.49	0.61	0.58	0.70
RCSVM	0.07	0.25	0.38	0.34	0.28
LP-PUL	0.51	0.65	**0.70**	**0.71**	**0.79**
NMFPUL	**0.689**	**0.699**	**0.705**	**0.716**	0.727

Table 10. F1 Score values for different algorithms on Document Collection SyskillWebert.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.51	0.73	0.71	0.80	0.81
PU-LP	0.58	0.68	0.82	0.73	0.70
RCSVM	0.40	0.39	0.37	0.34	0.30
LP-PUL	**0.69**	**0.80**	**0.88**	**0.90**	**0.89**
NMFPUL	0.613	0.633	0.647	0.670	0.688

Table 11. F1 Score values for different algorithms on Document Collection WAP.

Algorithm	# Labeled Documents				
	1	5	10	20	30
K-Means	0.32	0.46	0.49	0.43	0.40
PU-LP	0.18	0.39	0.41	0.43	0.38
RCSVM	0.02	0.14	0.27	0.34	0.41
LP-PUL	0.33	0.46	0.47	0.55	0.51
NMFPUL	**0.739**	**0.740**	**0.745**	**0.746**	**0.748**

Based on our analysis, our approach outperforms most baseline algorithms across most document collections. With few labeled documents, our algorithm achieves satisfactory results comparable to the performance of other techniques. Notably, our algorithm does not exhibit the best performance among the compared techniques in datasets with a smaller number of documents or classes. However, as we analyze larger datasets, *NMFPUL* consistently outperforms all the algorithms, particularly when the number of labeled positive documents is small. This observation aligns with the nature of *NMF*, as it tends to yield better results when applied to larger volumes of data [X. Lin 2020]. While *NMF* effectively reduces sparsity and noise in data when applied in its original form, it requires a minimum amount of data to deliver better performance.

6. Conclusion

This paper presents the adaptation of the algorithm Non-negative Matrix Factorization for application as a text classifier for Positive Unlabeled data, denominated *NMFPUL, NMF* for Positive Unlabeled Learning. The framework shown modifies the structure of *NMF* to better classify text documents to deal with the unlabeled data, which is a recurrent issue on data generated by the major applications. Using *NMF* to classify text data, which was already greatly used to reduce dimensionality and provide topics from documents and words, allows us to lay out a different look at how matrix factorization could support classification jobs with a reasonable performance. Lastly, the structure of our adapted *NMF* model provides a bridge to procure mathematical insight into PU data classification.

We conduct a substantial experiment comparing *NMFPUL* with the results from other algorithms and shows that our proposal could surpass some state-of-art methods for text classification in PU data. Our algorithm is tested along a different number of labeled examples and through extensive iterations and metrics method evaluation and provided performance better or close to the best methods of *PUL*.

In future work, we intend to apply the same methodology to other datasets in order to verify the consistency of the algorithm. Also, we aim to apply Deep Learning techniques to address PU data problems. One potential direction is to investigate using a Deep NMF model or other variants specifically tailored for text data classification. By leveraging the capabilities of Deep Learning, we anticipate the potential for improved performance and enhanced representation learning in PU data scenarios. This avenue of research holds promise for advancing the field and addressing the unique challenges posed by PU data classification.

References

Bekker, J. and Davis, J. (2018). Learning from positive and unlabeled data under the selected at random assumption. *Journal of Machine Learning Research*, 1.

Bekker, J. and Davis, J. (2020). Learning from positive and unlabeled data: a survey. Springer Nature 2020.

Bekker, J., Robberechts, P., and Davis, J. (2019). Beyond the selected completely at random assumption for learning from positive and unlabeled data. *Journal of Machine Learning Research*, 1.

Bian, P., Liu, L., and Penny, S. (2021). Detecting spam game reviews on steam with a semi-supervised approach. *Australian National University*, 06.

Carnevali, J. C., Geraldelli Rossi, R., Milios, E., and de Andrade Lopes, A. (2021). A graph-based approach for positive and unlabeled learning. *Information Sciences 580 (2021)*, 580.

Faleiros, T., Valejo, A., and de Andrade Lopes, A. (2020). Unsupervised learning of textual pattern based on propagation in bipartite graph. Intelligent Data Analysis.

He, D., Pan, M., Hong, K., Cheng, Y., Chan, S., Liu, X., and Guizani, N. (2020). Fake review detection based on pu learning and behavior density. *IEEE Network*, 92.

Hien, L. T. K. and Gillis, N. (2020). Algorithms for nonnegative matrix factorization with the kullback-leibler divergence. *Journal of Scientific Computing*, 87.

Jaemin, Y., Kim, J., Yoon, H., Kim, G., Jang, C., and U, K. (2022). Graph-based pu learning for binary and multiclass classification without class prior. *Knowledge and Information Systems (2022)*, 10.

Jaskie, K. and Spanias, A. (2019). Positive and unlabeled learning algorithms and applications: a survey. *SenSIP Center, School of ECEE*, 1.

Ji, Z., Du, C., Jiang, J., Zhao, L., Zhang, H., and Ganchev, I. (2023). Improving non-negative positive-unlabeled learning for news headline classification. *IEEE Access*, 11.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. Information.

Lee, D. and Seung, H. (2000). Algorithms for non-negative matrix factorization. Neural Inf. Process. Syst.

Li, M., Pan, S., Zhang, Y., and Cai, X. (2016). Classifying networked text data with positive and unlabeled examples. Pattern Recognition Letters.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Yang, L., and P. S. Yu, S. (2022). A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology.

Li, X. and L, B. (2003). Learning to classify texts using positive and unlabeled data. volume 1, pages 587–592.

Ma, S. and Zhang, R. (2017). Pu-lp: A novel approach for positive and unlabeled learning by label propagation. *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 01.

Mahesh, B. (2020). Machine learning algorithms - a review. *International Journal of Science and Research (IJSR)*, 9.

Naeem, M., Jamal, T., Diaz-Martinez, J., A. Butt, S.and Montesano, N., I. Tariq, M., De-la Hoz-Franco, E., and De-la Hoz-Valdiris, E. (2021). Trends and future perspective challenges in big data.

P. Tan, M. Steinbach, A. K. and Kumar, V. (2019). *Anomaly Detection*. Pearson.

Rossi, R. G., Marcacini, R. M., and Rezende, S. O. (2013). Benchmarking text collections for classication and clustering tasks. Technical report.

van Engelen, J. E. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Wang, Z., Jiang, J., and Long, G. (2022). Positive unlabeled learning by semi-supervised learning. *Australian Artificial Intelligence Institute,*, 213.

Wu, M., Pan, S., Du, L., and Zhu, X. (2021). Learning graph neural networks with positive and unlabeled nodes. *ACM Trans. Knowl. Discov*, 101.

Wu, Z., Cao, J., Wang, Y., Wang, Y., Zhang, L., and Wu, J. (2020). hpsd: A hybrid pu-learning-based spammer detection model for product reviews. *IEEE TRANSACTIONS ON CYBERNETICS*, 50.

X. Lin, P. C. B. (2020). Optimization and expansion of non-negative matrix factorization. BMC Bioinformatics.

Yang, F., Dragut, E., and Mukherjee, A. (2020). Claim verification under positive unlabeled learning. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 92.