# Random Walk in Feature–Sample Networks for Semi-Supervised Classification

Filipe Alves Neto Verri
Institute of Mathematical and Computer Sciences
University of São Paulo – São Carlos, SP, Brazil
School of Electrical, Computer and Energy Engineering
Arizona State University – Tempe, AZ, USA
E-mail: filipeneto@usp.br

Liang Zhao
Ribeirão Preto School of Philosophy, Science and Literature
University of São Paulo
Ribeirão Preto, SP, Brazil
E-mail: zhao@usp.br

***Abstract*—Positive-unlabeled learning is a semi-supervised task in which only some positive-labeled and many unlabeled samples are available. The goal of its transductive setting is to label all unlabeled data at once. In this paper, we developed a technique to grade positive-class pertinence levels of each sample, and we interpret the grades to classify the unlabeled ones. In our method, a sparse binary matrix represents the input data, which determines the feature–sample network whose vertices represent samples and attributes. The limiting probabilities of a random walk in the network estimate the pertinence levels. The results are evaluated regarding both class discrimination and classification accuracy. Computer simulations reveal that our model performs well in positive-unlabeled learning, especially with few labeled samples. Notably, the outcomes compare to the results from supervised methods, which profit from most data labeled. Additionally, the technique has linear time and space complexity if the input dataset is already in a sparse representation. The low computational cost of the construction and update of the feature–sample network allows for extensions of the technique to several learning problems, including online learning and dimensionality reduction.**

***Index Terms*—Semi-supervised classification, complex networks, positive-unlabeled learning, random walk.**

 Published version: `http://ieeexplore.ieee.org/document/7839592/`.
DOI 10.1109/BRACIS.2016.051.

## I. Introduction

Several machine learning techniques can learn not only from labeled data but also from unlabeled instances. Such models belong to the semi-supervised learning paradigm [1]. Partially labeled datasets are plentiful because more information is generated in our world than we can label by hand.

Many semi-supervised techniques model the input data either as graphs or as complex networks [2]–[5]. In both cases, each vertex usually represents a data sample, and an edge exists if its endpoints satisfy a predefined affinity rule. In graph-based techniques, an optimization process propagates the labels from labeled vertices to the unlabeled ones. In techniques based on complex networks, the algorithm classifies data by exploring topological and evolution properties of a certain collective dynamics. These methods provide a flexible and robust learning process.

A special scenario of semi-supervised classification comprises only few positive-labeled and many unlabeled samples; this problem is called positive-unlabeled (PU) learning [6]–[8]. The goal of PU learning tasks is either to build a classifier that discriminates positive and negative samples or to label all unlabeled input samples at once. Techniques that accomplish the former are inductive; while the ones that achieve the latter are transductive. Some graph-based techniques have been proposed [9], [10], but no complex-network approach exists.

We introduce a transductive PU learning technique based on complex networks with steps: *a)* The input dataset is converted into a sparse binary representation. (We use the terms feature and attribute interchangeably.) *b)* The binary representation models a network in which a vertex represents either a sample or an attribute. *c)* A random walk process is performed over the network taking into account the labeled samples; this process is a discrete Markov chain with states associated with the network's vertices. *d)* The positive-class pertinence level of each unlabeled sample is calculated from the limiting distribution of the Markov chain. *e)* Unlabeled samples are classified using the positive-class pertinence and with knowledge of the positive-class prior probability.

The model is evaluated regarding both class discrimination and classification accuracy. To assess how well it discriminates positive from negative samples, we introduce a measurement of class discrimination and a fair baseline algorithm to compare with. Concerning the classification accuracy, the results are compared with state-of-the-art techniques.

The proposed scheme excels on PU learning problems. Compared with PU classifiers, the technique surpasses other methods if the prior probabilities are known. Compared with supervised classifiers, the results are similar, even though those classifiers profit from much more labeled samples available during the training step. The research shows potential for a broad range of applications.

The rest of this paper is organized as follows. Sections II and III describe the proposed model and its computational complexity. In Section IV, computer simulations illustrate the learning process and assess its performance. Finally, Sections V and VI discuss and conclude this paper.

## II. Model Description

Let $\mathcal{D} = \{\vec{x}_1, \ldots, \vec{x}_N\}$ be the dataset where each data item $\vec{x}_i$ is either a *positive* or a *negative* sample. Examples $\vec{x}_i$ are positive labeled for all $i \in \mathcal{P}$, and the remaining samples are unlabeled. The prior probability $P^+$ of the positive class is known. Our goal is to estimate a positive-class pertinence level $f(\vec{x}_i)$ of each unlabeled data $\vec{x}_i$, $i \notin \mathcal{P}$, and to classify them using the priors.

In the next subsections, we explain the steps of our learning algorithm to solve the stated problem.

### A. Conversion to the Binary Sparse Representation

In our model, the input dataset must satisfy three requirements: each sample is a binary feature vector, an attribute with value 1 indicates the *presence* of a characteristic of that data instance, and the similarity of two samples depends on the number of shared characteristics but not on mismatching and absent features. Such requirements likely cause feature vectors to be sparse.

Most of the datasets are easily converted to this representation. We converted datasets composed of numerical and categorical attributes as follows: *a)* Replace each categorical feature $x \in \{X_1, \ldots, X_K\}$ of each sample by a binary feature vector $\{b_k\}_{k=1,\ldots,K}$ such that $b_k = 1 \iff x = X_k$. The sparsity of the new features is proportional to the number of possible categorical values $K$. *b)* Discretize numerical attributes disregarding the class information. The most common approaches are by equal interval width and by equal frequency. *c)* Convert the categorical features obtained from discretization to vectors of binary features as well.

### B. Construction of the Feature–Sample Network

We derive a complex network from the binary dataset. Let $\mathcal{B} = \{\vec{x}_1, \ldots, \vec{x}_N\}$ be the set where each element $\vec{x}_i$ is a binary feature vector $(x_{i1}, \ldots, x_{iM}) \in \{0,1\}^M$. The *Feature–Sample Network* $\mathcal{G}$ is the bipartite complex-network whose edges associates samples and features of the dataset $\mathcal{D}$. A simple, unweighted, undirected graph $(\mathcal{V}, \mathcal{E})$ represents such network. The vertex set $\mathcal{V}$ is $\{v_1, \ldots, v_N, v_{N+1}, \ldots, v_{N+M}\}$ and an edge exists between sample $v_i$ and feature $v_{N+j}$ if $x_{ij} = 1$. The adjacency matrix $A = (a_{ij})$ of $\mathcal{G}$ has elements

$$a_{ij} = a_{ji} = \begin{cases} x_{i,j-N} & \text{if } 1 \le i \le N \text{ and } N < j \le N+M, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We suppose that $\mathcal{G}$ has a single connected component. If it is false in an experiment, we consider only the largest connected component that contains all (or most) of the labeled samples.

### C. Modeling of the Random Walk Process

The next step is to perform a random walk over the network. This random process is a discrete Markov chain with transition matrix $P = (p_{ij})$, such that $p_{ij}$ is the probability of going from $v_i$ to $v_j$. We model $P$ to guarantee the existence and uniqueness of the limiting distribution of such Markov chain.

Since the $\mathcal{G}$ is connect, the process is a time-homogeneous and irreducible Markov chain. An irreducible Markov chain has a unique stationary distribution if and only if all states are positive recurrent [11]. Since $\mathcal{G}$ is undirected and finite, all states of a random walk over $\mathcal{G}$ are positive recurrent. Thus, a unique stochastic vector $\vec{\pi}$ exists such that

$$\vec{\pi} P = \vec{\pi}, \tag{2}$$

if $p_{ij} > 0$ for all $a_{ij} = 1$.

We also want to reach the stationary distribution $\vec{\pi}$ from any initial distribution. The limiting distribution of a random walk is reached independently of the initial conditions if the irreducible Markov chain is ergodic, that is, both aperiodic and positive recurrent [11]. Since every state of a bipartite graph has an even period, the limiting distribution of a random walker in $\mathcal{G}$ may never reach the stationary distribution. To achieve an ergodic Markov chain, we include non-zero entries on the main diagonal of $P$.

Finally, the transition probabilities are

$$p_{ij} = \frac{w_{ij}\nu_j}{\lambda \nu_i}, \tag{3}$$

where $W = (w_{ij})$ such that

$$w_{ij} = \begin{cases} 1 & \text{if } i = j, \\ \beta_i a_{ij} & \text{otherwise,} \end{cases} \tag{4}$$

and $\vec{\nu} = (\nu_1, \ldots, \nu_{N+M})$ is the eigenvector associated with the leading eigenvalue $\lambda$ of the matrix $W$. The scaling factor $\beta_i$ is $\beta > 1$ if $\vec{x}_i$ is a positive-labeled sample. Otherwise, the scaling factor is 1.

### D. Estimation of the Positive-class Pertinence Level

The transition matrix $P$ describes a system with the desired behavior: the limiting distribution $\vec{\pi}$ is reached independently of the initial setting; the states associated with features relevant to the positive class will have high stationary probabilities; and the limiting probabilities related to positive samples are expected to be greater than the ones associated with negative samples.

We estimate the positive-class pertinence level $f(\vec{x}_i) = \pi_i$ for all $i \notin \mathcal{P}$. The stochastic vector $\vec{\pi} = (\pi_i)$ is a eigenvector associated with the leading eigenvalue of the matrix $P^T$. Alternatively, it can be calculated by iterating the system $\vec{\pi}(t+1) = \vec{\pi}(t)P$ with any initial configuration.

### E. Classification of the unlabeled samples

We classify a unlabeled sample $\vec{x}_i$, $i \notin \mathcal{P}$, as positive if its pertinence level $f(\vec{x}_i)$ is greater than a certain threshold; which satisfies the expected number positive samples according to the prior probability $P^+$

The predicted class $c(\vec{x}_i)$ of an unlabeled sample $\vec{x}_i$ is

$$c(\vec{x}_i) = \begin{cases} +1 & \text{if } f(\vec{x}_i) > f^{\text{ordered}}_{\lceil (N - |\mathcal{P}|)P^+ \rceil} \\ -1 & \text{otherwise,} \end{cases} \tag{5}$$

where $f^{\text{ordered}}_n$ is the $n$-th greatest value of $f(\vec{x}_i)$ for all $i \notin \mathcal{P}$.

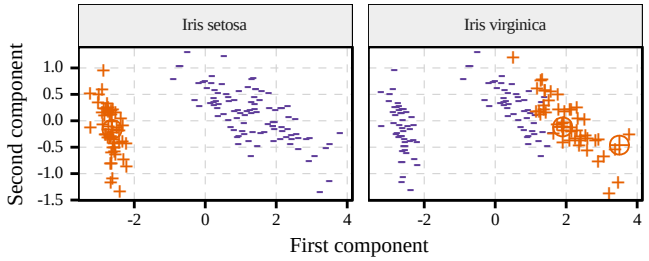


Figure 1. PCA projection of the Iris dataset, with samples of species *Iris setosa* (left-hand plot) and *Iris virginica* (right-hand plot) as the positive class.

## III. Time and Space Complexity

We analyze the computational complexity of our algorithm regarding $N$ and $D$, the number of samples and features.

The conversion of the input dataset into a sparse binary representation takes $\mathcal{O}(DN \log_2 N)$. The worst case scenario happens when all $D$ numerical attributes must be discretized. Since the features along all samples would need to be sorted, the average sorting time adds up to the time complexity.

The construction of the feature–sample network takes $\mathcal{O}(DN)$, since it highly sparse, and the number of edges is *limited* by $DN$.

Both modeling the process and searching the stationary distribution take $\mathcal{O}(DN)$. The two depend on the choice of the eigenvalue algorithm and the matrix representation. With sparse matrices and iterative algorithms – for example, the power iteration algorithm – the time and space requirement is proportional to $DN$.

In summary, our method runs in $\mathcal{O}(DN)$ if the input dataset is binary, and in $\mathcal{O}(DN \log_2 N)$ otherwise. Since we only store matrices with up to $DN$ nonzero entries, our method has space complexity $\mathcal{O}(DN)$.

## IV. Experimental Simulations

In the following subsections, we provide computer simulations to illustrate the learning process and assess its performance in real-world datasets.

### A. Illustrative Example

The UCI Iris dataset [12] comprises 50 samples for each of three species of Iris flower—*Iris setosa*, *Iris virginica*, and *Iris versicolor*. Each data item contains 4 numerical features which stand for measurements of the width and length of the sepals and petals in centimeters.

Similarities between samples of the three classes vary considerably. *Iris setosa* samples are quite distinct from the other two species. *Iris virginica* and *Iris versicolor* samples, on the other hand, share more similarities between themselves. Figure 1 clarify this property, showing the first two principal components of the PCA projection. We consider two input setups: *a) Iris setosa* as the positive class and only a single positive sample labeled; and *b) Iris virginica* as the positive class and two positive samples labeled. The labeled samples are circled in Figure 1. For both scenarios, we describe our learning process step by step.

Table I
Features of labeled samples shown in Figure 1.

| Class | Sepal Length | Sepal Width | Petal Length | Petal Width |
|---|---|---|---|---|
| *setosa* | 5.0 | 3.4 | 1.5 | 0.2 |
| *virginica* | 7.7 | 2.8 | 6.7 | 2.0 |
| *virginica* | 6.2 | 3.4 | 5.4 | 2.3 |

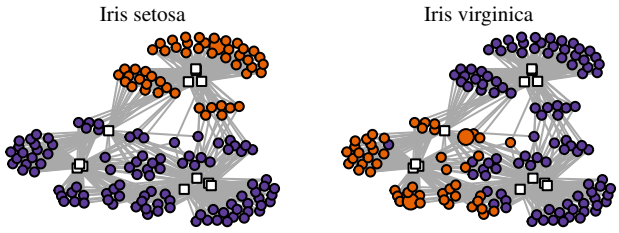


Figure 2. Feature–sample network for Iris dataset with samples of species *Iris setosa* (left-hand side) and *Iris virginica* (right-hand side) as the positive class. Circles are vertices associated with samples and squares with features. Positive samples are highlighted in orange (light).

*1) Binary Sparse Representation:* We convert the dataset into a sparse binary representation; independently of the labeled samples. Numerical attributes are discretized in 3 intervals by frequency.

The discretized representation reduce the numerical detail while holding sufficient information for the classification task. Tables I and II compare both representations. The first line represent *Iris setosa* samples, and the remaining represent the *Iris virginica* samples. The discretization intervals are below the features names in Table II. Although the discrete representation loses information, the sophistication of the learning process overcomes its simplification.

*2) Feature–Sample Network:* With the binary representation of data and Equation (1), we construct the feature–sample network. Figure 2 illustrates two networks for the Iris dataset. Circles are vertices of *samples*, and squares are vertices of *features*. The left-hand network, in which *Iris setosa* is the positive class, has positive samples in light orange, and the labeled samples are bigger. The right-hand network, in which *Iris virginica* is the positive class, has the same characteristics.

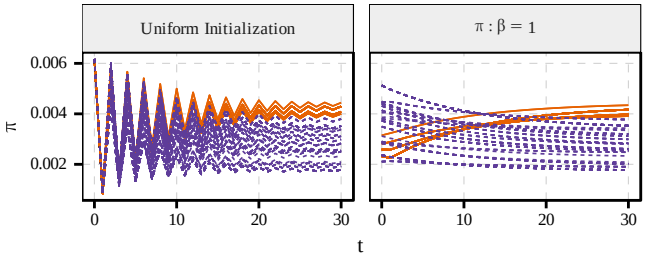


Figure 3. Evolution of the probability distribution with two different initial distribution. The process is modeled from Iris dataset with one labeled sample of class *Iris setosa* and $\beta = 6$. Only values associated with unlabeled samples are shown. Solid orange lines and dashed purple lines are associated with positive and negative samples in that order.

Table II
SPARSE BINARY REPRESENTATION OF THE SAMPLES IN TABLE I.

| Sepal Length ∈ | | | Sepal Width ∈ | | | Petal Length ∈ | | | Petal Width ∈ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $[4.3, 5.5)$ | $[5.5, 6.4)$ | $[6.4, 7.9]$ | $[2.0, 3.0)$ | $[3.0, 3.3)$ | $[3.3, 4.4]$ | $[1.0, 3.0)$ | $[3.0, 5.0)$ | $[5.0, 6.9]$ | $[0.1, 1.0)$ | $[1.0, 1.7)$ | $[1.7, 2.5]$ |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

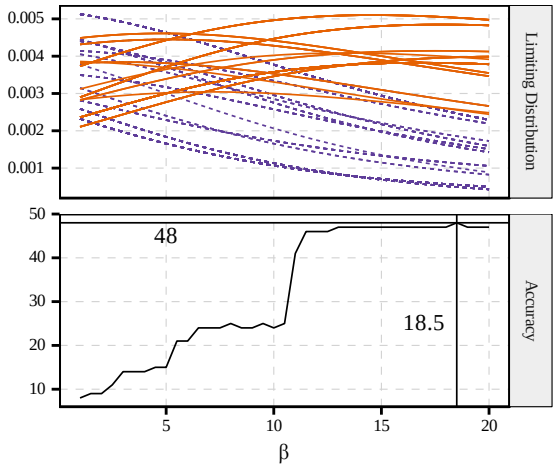


Figure 4. (Top) Limiting distribution of the process modeled from the Iris dataset with two labeled sample of class *Iris virginica* for varying $\beta$. Only values associated with unlabeled samples are shown. Solid orange lines and dashed purple lines are associated with positive and negative samples in that order. (Bottom) Accuracy for varying $\beta$, which the former is the number of positive samples among the 50 samples with greatest scores.

*3) Random Walk:* With the network and Equation (3), we compute the transition matrix $P$. For one labeled sample of species *Iris setosa*, we find the stationary distribution $\vec{\pi}$ iteratively, with $\beta = 6$. Independently of the initial distribution, we should obtain the same results. Figure 3 shows the evolution of $\pi_i(t)$, $i$ such that $\vec{x}_i$ is unlabeled, for two initial distributions $\vec{\pi}(0)$: the uniform distribution, on the left-hand plot, and the limiting distribution of a process $P$ given that $\beta = 1$, that is, ignoring the labeled sample.

The probability of a random walker be in a state of unlabeled positive samples (solid orange lines) surpasses the same on negative samples (dashed purple lines). This behavior holds for both initial distributions and is guaranteed independently of the initial configuration. Starting with the limiting distribution ignoring the labeled sample, the limiting probabilities associated with positive samples clearly increase over time.

We also study the learning process for the second scenario. For $\beta = 1, 1.5, \ldots, 20$, Figure 4 shows the limiting distribution and the accuracy, which is the number of samples of class *Iris virginica* among the 50 samples with greatest scores $f(\vec{x})$. Differently from the former scenario, for small $\beta$ the probability of a random walker be in a state associated with positive samples (solid orange lines) is not consistently superior to the same on negative samples (dashed purple line). However, for $\beta > 11$, more than 40 out of 50 samples are from the *Iris virginica* class. With $\beta = 18.5$, only two samples are misclassified.

Table III
UCI DATASETS ALONG THE CLASS CHOSEN TO BE THE POSITIVE ONE. $P^+$ IS THE PROPORTION OF SAMPLES IN THE INDICATED CLASS.

| Dataset | Positive class | $P^+$ |
|---|---|---|
| Breast 2010 | adi | 0.21 |
| Ecoli | cp | 0.43 |
| Glass | building windows non float processed | 0.36 |
| Iris | setosa, versicolor, virginica | 0.33 |
| Wine | two | 0.40 |

### B. Performance Comparison

We compare our model with a baseline semi-supervised method. Such baseline method is based on the neighborhood graph. Both approaches not only rely on the same assumptions for the input dataset but also use the same classification mechanism; providing a fair comparison of the positive-class pertinence levels.

*1) Learning from the Neighborhood Graph:* The $k$-NN graph of a dataset $\mathcal{D} = \{\vec{x}_1, \ldots, \vec{x}_N\}$ is a graph where each vertex $v_i$, associated with sample $\vec{x}_i$, connects with all vertices $v_j$ such that $\vec{x}_j$ is within the $k$-neighborhood of $\vec{x}_i$. We use the asymmetric binary similarity—proportional to the number of attributes that are present in both samples—to calculate the neighborhoods.

The positive-class pertinence level $f^{k\text{-NN}}(\vec{x}_i)$, for all $i \notin \mathcal{P}$, is $(\min_{j \in \mathcal{P}} l_{ij})^{-1}$, where $l_{ij}$ is the length of the shortest path between vertices $v_i$ and $v_j$ in the $k$-NN graph. In other words, the positive class pertinence of a sample is inversely proportional to the shortest distance from the associated vertex to any labeled vertex.

*2) Benchmark Datasets:* Five UCI datasets [12] are used to compare our model with the $k$-NN method. In each dataset, the largest class is the positive one. In the Iris dataset, however, the classes are of the same size, and all three possible cases were considered. Table III presents the datasets along with the proportion $P^+$ of positive samples; which we use as the positive-class prior probability.

*3) Separateness:* The *separateness* evaluates how well a model differentiates positive from negative samples. Given the positive pertinence $f(\vec{x}_i)$ for all unlabeled samples $\vec{x}_i$, the separateness is

$$\frac{1}{N - |\mathcal{P}|} \sum_{i \notin \mathcal{P}} \tilde{f}(\vec{x}_i) \delta(\vec{x}_i) \tag{6}$$

where $\tilde{f}$ is the pertinence level of unlabeled samples normalized with standard score, and $\delta(\vec{x}_i)$ is $+1$ if $\vec{x}_i$ is positive or

Table IV
SEPARATENESS OF THE METHODS ON THE UCI DATASETS. FOR EACH SETTING, THE BEST PARAMETER COMBINATION IS SHOWN.

| Dataset | Our method | ($\beta$, $m$) | $k$-NN graph | ($k$, $m$) |
|---|---|---|---|---|
| | **1% labeled** | | | |
| Breast 2010 | $\mathbf{0.60 \pm 0.11}$ | (10, 5) | $0.53 \pm 0.13$ | (20, 4) |
| Ecoli | $\mathbf{0.60 \pm 0.12}$ | (50, 3) | $0.56 \pm 0.07$ | (15, 4) |
| Glass | $0.12 \pm 0.14$ | (50, 4) | $\mathbf{0.15 \pm 0.14}$ | (15, 4) |
| Iris (setosa) | $\mathbf{0.91 \pm 0.01}$ | (20, 3) | $0.75 \pm 0.05$ | (20, 5) |
| Iris (versicolor) | $\mathbf{0.77 \pm 0.04}$ | (20, 3) | $0.58 \pm 0.08$ | (20, 3) |
| Iris (virginica) | $\mathbf{0.65 \pm 0.01}$ | (35, 2) | $0.53 \pm 0.29$ | (20, 3) |
| Wine | $\mathbf{0.61 \pm 0.06}$ | (5, 5) | $0.40 \pm 0.19$ | (7, 4) |
| | **10% labeled** | | | |
| Breast 2010 | $\mathbf{0.65 \pm 0.04}$ | (15, 5) | $0.54 \pm 0.09$ | (19, 4) |
| Ecoli | $\mathbf{0.78 \pm 0.02}$ | (50, 3) | $0.78 \pm 0.04$ | (19, 4) |
| Glass | $0.23 \pm 0.08$ | (50, 3) | $\mathbf{0.25 \pm 0.07}$ | (14, 4) |
| Iris (setosa) | $\mathbf{0.91 \pm 0.00}$ | (5, 3) | $0.81 \pm 0.04$ | (20, 5) |
| Iris (versicolor) | $\mathbf{0.81 \pm 0.03}$ | (50, 3) | $0.68 \pm 0.14$ | (15, 3) |
| Iris (virginica) | $\mathbf{0.79 \pm 0.03}$ | (35, 3) | $0.54 \pm 0.13$ | (17, 3) |
| Wine | $\mathbf{0.77 \pm 0.04}$ | (20, 3) | $0.55 \pm 0.15$ | (17, 4) |

Table V
ACCURACY OF THE METHODS ON THE UCI DATASETS. FOR EACH SETTING, THE BEST PARAMETER COMBINATION IS SHOWN.

| Dataset | Our method | ($\beta$, $m$) | $k$-NN graph | ($k$, $m$) |
|---|---|---|---|---|
| | **1% labeled** | | | |
| Breast 2010 | $\mathbf{0.91 \pm 0.07}$ | (5, 5) | $0.88 \pm 0.05$ | (20, 4) |
| Ecoli | $0.87 \pm 0.02$ | (5, 3) | $\mathbf{1.00 \pm 0.00}$ | (1, 2) |
| Glass | $\mathbf{0.58 \pm 0.07}$ | (50, 2) | $0.57 \pm 0.08$ | (20, 4) |
| Iris (setosa) | $\mathbf{1.00 \pm 0.00}$ | (10, 3) | $\mathbf{1.00 \pm 0.00}$ | (1, 2) |
| Iris (versicolor) | $\mathbf{0.90 \pm 0.03}$ | (40, 3) | $0.80 \pm 0.14$ | (16, 5) |
| Iris (virginica) | $\mathbf{0.84 \pm 0.15}$ | (30, 3) | $0.81 \pm 0.18$ | (20, 3) |
| Wine | $\mathbf{0.78 \pm 0.03}$ | (5, 5) | $0.71 \pm 0.11$ | (14, 3) |
| | **10% labeled** | | | |
| Breast 2010 | $\mathbf{0.95 \pm 0.02}$ | (15, 5) | $0.86 \pm 0.05$ | (12, 5) |
| Ecoli | $0.91 \pm 0.02$ | (10, 3) | $\mathbf{1.00 \pm 0.00}$ | (1, 3) |
| Glass | $0.64 \pm 0.05$ | (50, 3) | $\mathbf{0.66 \pm 0.05}$ | (13, 4) |
| Iris (setosa) | $\mathbf{1.00 \pm 0.00}$ | (5, 3) | $\mathbf{1.00 \pm 0.00}$ | (1, 2) |
| Iris (versicolor) | $\mathbf{0.93 \pm 0.02}$ | (40, 3) | $0.91 \pm 0.03$ | (20, 4) |
| Iris (virginica) | $\mathbf{0.95 \pm 0.03}$ | (20, 3) | $0.76 \pm 0.06$ | (14, 3) |
| Wine | $\mathbf{0.91 \pm 0.03}$ | (35, 4) | $0.79 \pm 0.07$ | (18, 4) |

$-1$ otherwise. The standard score normalization subtracts all pertinence levels $f$ by the average value and divides by the standard deviation.

The separateness is positive when the majority of positive-labeled samples has score above the average. Negative values or near zero indicate failure to distinguish between positive and negative classes.

*4) Discrimination Results:* To evaluate our technique, the binary representation was created with numerical attributes discretized in $m = 2, 3, 4, 5$ intervals. The parameter $k$ ranges from 1 to 20, and the parameter $\beta$ in $\{5, 10, \ldots, 50\}$. The number of initially labeled positive samples are 1% or 10% of positive samples.

Our method separates better in seven out of eight settings. The results are independent of the number of labeled samples, which are listed in Table IV along with the $k$-NN's results. The table shows the mean separateness and the standard deviation for 20 independent sets of labeled items, with the parameters that yield the best performance. The proposed technique present relative low standard deviation, implying the model is less sensible to the choice of labeled samples.

*5) Classification Results:* Excellent accuracy is observed in our method for few labeled samples and the naive classification mechanism. Using the predicted class $c$ and the same parameter settings, Table V shows the accuracies for the UCI datasets. Surprisingly, the $k$-NN method obtained better accuracy for the Ecoli dataset besides its worse separateness. Our method's results compare to those acquired by recently proposed techniques that profit from more than twice of labeled samples [13].

### C. Document Classification

We also perform tests on the Reuters-21578 ApteMod dataset. This dataset is a collection of 10,788 documents from the Reuters financial newswire service. Each document belongs to one or more of the 90 categories.

Our method is compared with state-of-the-art algorithms studied in [14]. All methods are adaptations of support vector machines. For each of the top 10 most populated categories, we derive a learning scenario where documents that belong to that category are treated as positive. We label 20% of the positive samples using a biased labeling process, which labels highly correlated documents; refer to [14] for more details.

Table VI
AVERAGE F-SCORE ON THE REUTERS DATASET WITH DIFFERENT CLASSES AS POSITIVE. FOR EACH SETTING, THE BEST NUMBER OF SELECTED FEATURES $D$ IS SHOWN. COMPARATIVE RESULTS OBTAINED FROM [14].

| Positive class | Our method ($D$) | EN D | SNOB MC D | TBSVM |
|---|---|---|---|---|
| Earn | 0.431 (28) | 0.573 | **0.575** | 0.536 |
| Acq | 0.347 (23) | 0.445 | **0.495** | 0.431 |
| Money-fx | **0.243** (8) | 0.188 | 0.215 | 0.174 |
| Grain | 0.228 (4) | 0.190 | **0.233** | 0.166 |
| Crude | **0.290** (4) | 0.191 | 0.226 | 0.172 |
| Trade | **0.213** (12) | 0.180 | 0.195 | 0.162 |
| Interest | **0.249** (5) | 0.135 | 0.152 | 0.133 |
| Ship | **0.274** (5) | 0.116 | 0.143 | 0.105 |
| Wheat | **0.441** (3) | 0.122 | 0.144 | 0.113 |
| Corn | **0.256** (2) | 0.097 | 0.108 | 0.084 |
| Average | **0.267** (5) | 0.224 | 0.249 | 0.208 |

This learning setting represents real PU learning tasks excellently. The biased labeling process reproduces the most common scenario where the positive examples are labeled based on search queries – for example, searches filtered by keywords – rather than uniformly at random [14].

For each document, we tokenize the text in words and stem each word, obtaining 21,173 different word stems. The binary representation of the dataset is straightforward: each stem is a feature, and a sample document has value 1 for a stem if it occurs in the document. Since there are many features, we pre-select only the $D$ most frequent stems in the set of labeled positive samples. For all experiments, we fix $\beta = 10^5$; which has been found empirically for best results.

Table VI presents the classification results. Each value is the average F-score from 5 independent labeled seeds; except for our method, where we consider 20 independent random seeds. For each topic, we indicate the best number of selected features $D$.

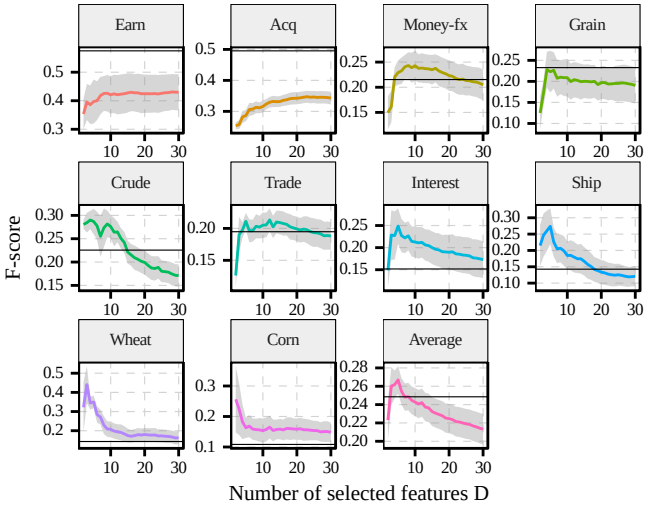


Figure 5. Average F-score and 95% confidence interval obtained by our method over the number of selected features $D$. The solid horizontal lines indicate the results of the state-of-the-art technique SNOB MC Double [14].

In Figure 5, one can observe the F-score average and the 95% confidence interval obtained for our method over the number of selected features. The horizontal lines indicate the state-of-the-art results.

Our model has the best results for almost all categories, except for the two most frequent topics. In average, our technique also outperforms the other methods. Both the number of selected features and the prior probabilities play a crucial role in the process. In future works, we shall conduct studies to estimate their optimal values.

## V. Discussion

The proposed model is simple to understand, to implement, and efficient to solve PU learning tasks with few or several labeled samples. Moreover, if the input dataset comes in a sparse representation, the computational complexity is linear. Nonetheless, three limitations will be addressed in the future.

Separateness is unsuitable for imbalanced datasets because it normalizes the class pertinence towards the average. A measurement that normalizes towards an estimated prior probabilities might provide better results.

The optimal $\beta$ is nontrivial, and a lower bound estimative that results in greater limiting probabilities for labeled samples should be possible.

The classification step depends on the assumption of knowing the positive-class prior. Such information may not feature in real datasets.

We can also extend our technique to several other learning problems: *a)* An alternative method for multi-class datasets can be obtained with competition dynamics instead of random walking [5]. *b)* Since addition and removal of samples in the feature–sample network and updates on the eigenvalues and eigenvectors cost little resources, the method can be adapted to deal with concept drift, outlier detection, classification on data streams, and active learning. *c)* An analysis of the limiting probabilities for the states associated with features would lead to new techniques for feature selection and dimensionality reduction.

## VI. Conclusion

We presented a PU learning system for transductive classification. We map the input data into a sparse binary representation and, afterward, into a complex network whose vertices represent samples and attributes. From only positive-labeled and unlabeled instances, we model a Markov Chain that outputs positive-class pertinence levels for the unlabeled samples. Knowing the priori probability of the positive class, we classify the unlabeled samples.

The model is illustrated step by step and evaluated against a baseline method and other techniques in literature. The proposed scheme offers high performance on PU learning problems, even with few labeled samples. However, two main limitations can impair its use on real applications: the estimation of the parameter $\beta$ and the class priors.

Once we address these issues, we can further explore the feature–sample network, generalize and extend the technique to other learning problems.

## Acknowledgments

This research was supported by the São Paulo State Research Foundation (FAPESP) and the Brazilian National Research Council (CNPq).

## References

[1] X. Zhu and A. B. Goldberg, "Introduction to Semi-Supervised Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[2] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 55–67, 2008.

[3] K. Zhang, L. Lan, J. T. Kwok, S. Vucetic, and B. Parvin, "Scaling Up Graph-Based Semisupervised Learning via Prototype Vector Machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 444–457, 2015.

[4] T. C. Silva and L. Zhao, *Machine Learning in Complex Networks*, 1st ed. New York, NY, USA: Springer, 2016.

[5] F. A. N. Verri, P. R. Urio, and L. Zhao, "Network Unfolding Map by Edge Dynamics Modeling," 2016, arXiv:1603.01182.

[6] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[7] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *International Joint Conference on Artificial Intelligence Proc.*, 2003, pp. 587–592.

[8] J. Mũnoz Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised One-Class Support Vector Machines for Classification of Remote Sensing Data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 8, pp. 3188–3197, 2010.

[9] S. Yu and C. Li, "PE-PUC: A Graph Based PU-Learning Approach for Text Classification," in *Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2007, vol. 4571, pp. 574–584.

[10] S. Segui, L. Igual, and J. Vitria, "Weighted Bagging for Graph Based One-Class Classifiers," in *Multiple Classifier Systems Proc.*, 2010, vol. 5997, pp. 1–10.

[11] S. M. Ross, *Introduction to Probability Models*, 11st ed. Orlando, FL, USA: Academic Press, 2014.

[12] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[13] L. Livi, A. Sadeghian, and W. Pedrycz, "Entropic One-Class Classifiers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3187–3200, 2015.

[14] N. Youngs, D. Shasha, and R. Bonneau, “Positive-Unlabeled Learning in the Face of Labeling Bias,” in *IEEE International Conference on Data Mining Workshop Proc.*, 2015, pp. 639–645.