View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Biblioteca Digital da Produção Intelectual da Universidade de São Paulo (BDPI/USP)

SISTEMA INTEGRADO DE BIBLIOTECAS
UNIVERSIDADE DE SÃO PAULO

Universidade de São Paulo

Biblioteca Digital da Produção Intelectual - BDPI

Departamento de Ciências de Computação - ICMC/SCC

Artigos e Materiais de Revistas Científicas - ICMC/SCC

2014-02

Unsupervised instance selection from text streams

Journal of Information and Data Management - JIDM, Porto Alegre, v.5, n.1, p.114-123, 2014
http://www.producao.usp.br/handle/BDPI/46577

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo

Unsupervised Instance Selection from Text Streams

Rafael Bonin[1], Ricardo M. Marcacini[2], Solange O. Rezende[1]

[1] Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP), São Carlos-SP, Brasil
rafabonin@grad.icmc.usp.br , solange@icmc.usp.br
[2] Universidade Federal do Mato Grosso do Sul (UFMS), CPTL, Três Lagoas-MS, Brasil
ricardo.marcacini@ufms.br

Abstract. Instance selection techniques have received great attention in the literature, since they are very useful to identify a subset of instances (textual documents) that adequately represents the knowledge embedded in the entire text database. Most of the instance selection techniques are supervised, i.e., requires a labeled data set to define, with the help of classifiers, the separation boundaries of the data. However, manual labeling of the instances requires an intense human effort that is impractical when dealing with text streams. In this article, we present an approach for unsupervised instance selection from text streams. In our approach, text clustering methods are used to define the separation boundaries, thereby separating regions of high data density. The most representative instances of each cluster, which are the centers of high-density regions, are selected to represent a portion of the data. A well-known algorithm for data sampling from streams, known as *Reservoir Sampling*, has been adapted to incorporate the unsupervised instance selection. We carried out an experimental evaluations using three benchmarking text collections and the reported experimental results show that the proposed approach significantly increases the quality of a knowledge extraction task by using more representative instances.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Instance Selection, Text Streams, Data Clustering

1. INTRODUCTION

The popularization of online platforms for publishing textual content has enabled a significant increase in the volume of data stored in text databases [Pang-Ning et al. 2006; Aggarwal 2012]. Moreover, besides the large volume of data, the frequent content updating of these databases becomes a key challenge for methods of knowledge extraction from texts, also referred in the literature as mining text streams [Gama 2010; Aggarwal 2012]. A promising way to address this challenge is through instance selection techniques [Liu 2010]. In these techniques, the goal is to identify a subset of instances (text documents) that adequately represents the existing knowledge in the entire text database.

Instance selection techniques allow the identification of text documents that are in the center regions of each category of documents, thereby representing the various topics from text database. Most existing instance selection techniques are supervised, i.e., require a labeled data set to define the boundaries among regions of different categories of documents [Liu and Motoda 2002; Reinartz 2002; Liu 2010]. However, the manual labeling of the text database requires an intense human effort that, in general, is not feasible in scenarios involving large databases and text streams. On the other hand, text clustering methods can be used to perform unsupervised instance selection. Clustering methods organize the instances into a number of clusters, where instances within the same cluster are closer to each other than to instances allocated in different clusters [Czarnowski 2012]. The clustering solution

Our research is supported by the FAPESP - São Paulo Research Foundation (grant 2010/20564-8), CAPES, and CNPq.
Copyright©2014 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

is then used to define the boundaries between regions, in general, separating regions of high data density [Liu 2010]. However, although the use of clustering methods allow instance selection without the need of a labeled data set, there still remains the challenge of dealing with text streams.

In this article, we present an approach for unsupervised instance selection from text streams. In particular, we consider a text stream that is monitored for a certain period of time, common to many real world applications. In this case, it is not possible to know a priori the number of instances that will be collected during the monitored period. In these scenarios, the *Reservoir Sampling* algorithm is widely used in order to define a subset of documents through data sampling [Gama 2012]. Besides the low computational cost, the *Reservoir Sampling* algorithm has the ability to perform a uniform random sampling of m elements without knowing the size of the data set. We incorporate an instance selection technique based on a text clustering method into a *Reservoir Sampling* algorithm, thereby allowing unsupervised instance selection from text streams. We carried out an experimental evaluation with three benchmarking text databases and the statistical analysis of the results indicates that our proposed approach is superior in terms of quality of the selected instances, when compared with the traditional *Reservoir Sampling.* Furthermore, we performed a thorough experimental analysis of the parameters involved in clustering algorithms, such as the number of clusters and number of selected instances, since this is an underexplored aspect in the instance selection literature.

2. BACKGROUND

In the context of this work, the main goal of the instance selection is to select a representative subset D_S from a textual dataset D_T, in which the performance P of a particular machine learning algorithm is maintained, i.e., $P(D_T) \cong P(D_S)$ [Reinartz 2002; Olvera-Lopez et al. 2010; Leyva et al. 2013]. Thus, instance selection can be defined as a data reduction technique that aims to reduce the computational task of extracting knowledge from text databases. According to Liu [2010], instance selection has the following prominent functions:

Enabling. Algorithms for knowledge extraction from texts are somewhat limited by their ability to handle large text databases. When a text database is very large, it is usually not possible to run an algorithm for extracting knowledge in a timely manner. Instance selection reduces the amount of data and then enables the handling of large databases without significant loss of the extracted knowledge.

Cleaning. The success of the knowledge extraction tasks depends on the quality of the textual databases. The well-known GIGO (Garbage In Garbage Out) principle is an example of the need to remove noise or irrelevant data from the texts. Instance selection aims at removing the noise and outliers from the database, thereby reducing the complexity of the problem by using only the (statistically) relevant instances of the domain.

Focusing. In real situations, textual data are collected and stored, usually including almost all kinds of information about the problem domain. However, many applications are usually related to only a few aspects of the problem domain. Therefore, it is naturally more efficient to select and focus on the data more related to the application domain, thereby selecting only the most representative instances according to some predefined criteria.

It is important to note that an instance selection technique must contain at least one of these three functions. Moreover, in some situations, there may be intersections between these functions. For example, the Cleaning function can be a direct consequence of Enabling and Focusing [Liu 2010].

Considering a scenario based on static data, the basic idea of instance selection techniques is to identify instances located in the center of one or more regions of the data boundaries. The boundaries define the regions of correlated data, which have a greater chance to form clusters or categories in the texts. In the presence of labeled data, such regions can be computed for each class label by training a classifier [Olvera-Lopez et al. 2010]. Similarly, regions and respective boundaries can be obtained in an unsupervised manner by using clustering methods [Liu 2010]. Once defined the boundaries, the instances that satisfy the minimum proximity to the center of the region can be selected, thereby discarding instances located at the margin of separation boundaries.

A current challenge is to perform the instance selection from text streams (non static data). In this scenario, there is usually no labeled data to compute the separation boundaries. Moreover, the text dataset is constantly updated. In fact, it is computationally expensive to repeat the instance selection process whenever there are significant changes in the database. Thus, in most real applications, the instance selection from text streams is based on random sampling techniques, in which each new instance has a certain probability of being selected. Instance selection based on random sampling contains the *Enabling* function because they allow to reduce the volume of data; and the *Cleaning* function, since noisy instances have a low probability of being selected at random (the expectation is that the number of noisy instances is small compared to the total number of instances). However , the use of random sampling does not guarantee a search for representatives instances, i.e., the *Focusing* function is not present.

3. UNSUPERVISED INSTANCE SELECTION BASED ON *RESERVOIR* SAMPLING

In this article, we incorporate the *Focusing* function in an instance selection process based on random sampling. We adapt the well-known *Reservoir Sampling* [McLeod and Bellhouse 1983; Vitter 1985; Gama 2012] algorithm by integrating a text clustering method to identify representative instances. This is an underexplored aspect in the literature, thereby allowing unsupervised instance selection from text streams without significant increase in computational cost.

The *Reservoir Sampling* algorithm aims to select a sample of instances of size m from a given text stream $S = \{d_1, d_2, ...\}$ of unknown size (greater than m), in which each instance has the same chance of being selected (uniform random sampling). In general, the size m is defined according to the memory restrictions involved in the application. In the *Reservoir Sampling* algorithm (Algorithm 1) the first m instances are read and inserted into a reservoir of size m. The next instances are read sequentially from the text stream S.

The probability of each new instance i, with $i > m$, be inserted in the reservoir is $\frac{m}{i}$, i.e., the ratio between the reservoir size and the number of instances so far obtained. It is important to note that for an instance i to be inserted in the reservoir, another instance in the reservoir must be removed. The probability of an instance i to be removed is given by the chance to be chosen randomly out of the reservoir $(\frac{1}{m})$ multiplied by the chance of the instance have been previously inserted in the reservoir $(\frac{m}{i})$, i.e., $\frac{1}{m} \times \frac{m}{i} = \frac{1}{i}$. Considering these probabilities, McLeod and Bellhouse [1983] demonstrated that the *Reservoir Sampling* algorithm results in a uniform random sampling for all possible instance sets of size m.

Random sampling can be effective for instance selection from text streams, considering the Enabling and Cleaning functions. However, the process can be improved significantly by incorporating the Focusing function using a criterion to identify the most representative instances. In our proposed approach, we use a technique based on clustering methods to obtain a subset of representative instances in order to present to the *Reservoir Sampling.* For this purpose, we define an additional buffer with the same size of reservoir for storing m instances from the text stream S. When the buffer is full, then a partitioning clustering algorithm is employed to obtain k clusters from instances of the buffer. The process for selecting the most representative instances is given by the identification of p instances closest to the cluster centroids, where p indicates a fraction in the range $(0, 1)$ of instances that will be presented to the algorithm *Reservoir Sampling.* For example, if $p = 0.1$ then 10% of the most representative instances are selected. In the proposed approach, the cluster centroids are calculated as the mean vector of all instances belonging to the cluster. This process is repeated every time the buffer is full and will terminate when there are no instances in the text stream, i.e., when the period

Algorithm 1: Algorithm *Reservoir Sampling* (adapted from Monahan [2011])

```
Input:
        $S = \{d_1, d_2, ...\}$: text stream
        $m$: reservoir size
initializing an array $R[]$ of size $m$ (reservoir)
for each new instance $i$ from $S$ do
    if $i \leq m$ then
        $R[i] \leftarrow S[i]$
    else
        $j \leftarrow$ random integer between $[1, i]$
        if $j \leq m$ then
            $R[j] \leftarrow S[i]$
        end
    end
end
return $R[]$
```

of time defined by the user has finished.

The key idea of our proposed approach is that noisy instances can be removed during sampling and at the same time enable uniform sampling among the most representative instances of the problem. Thus, while the traditional *Reservoir Sampling* presents the Cleaning and Enabling functions, we explore the use of clustering to select representative instances from a text stream as a promising way to incorporate Focusing function into Reservoir Sampling technique. The size m of the reservoir (and buffer), the number of clusters k and the fraction p of instances selected per cluster are parameters of the proposed approach.

Regarding the time complexity of the proposed approach for unsupervised instance selection, it is important to note that the partitioning clustering method (such as k-means [MacQueen 1967]) has complexity $O(km)$, where k is the number of clusters and m is the number of instances of the buffer. Selecting a fraction p of instances from clustering solution also has complexity $O(km)$. The clustering method is repeated $\frac{|S|}{m}$ times, where $|S|$ is the estimated number of instances collected from the text stream. Thus, the time complexity of the process can be defined as $O(\frac{k.m.|S|}{m})$. Considering the dominant variables of the problem, and assuming $k << |S|$, the time complexity of the proposed approach is linear $O(k.|S|)$, being competitive for many real applications.

4. EXPERIMENTAL EVALUATION

We carried out an experimental evaluation to assess the quality of the unsupervised instance selection from text streams, proposed in this article, using three benchmarking text databases. Table I presents the details of the text databases. Each database contains reference categories that are used as ground truth partitions for the result analysis.

Table I. Details of text databases used in the experimental evaluation.

Database	Source	#Features	#Instances	#Categories
20ng	Newsgroups Messages	18.745	18.828	20
NSF	National Science Foundation	10.160	10.521	16
Re8	Reuters-21578	7.555	7.674	8

The smaller dataset contains 7,674 documents while the larger dataset contains 18,828 documents. These datasets have been used in other studies on hierarchical clustering of documents [Zhao et al. 2005]. The 20ng dataset consists of e-mail messages organized into 20 mailing lists groups [Rennie 2008]. This collection is often used in text clustering tasks. The NSF dataset consists of public documents which describe scientific projects submitted to the *National Science Foundation (USA)* [1] [Pazzani and Meyers 2003]. Each document consists of a title and a short abstract of 150 words. The NSF dataset are organized into 16 categories representing research areas. The Re8 dataset consists of news reports extracted from Reuters-21578 dataset [Lewis 1997] and is provided by the *CSMining Group* [Pang 2010]. A more detailed description of these datasets is available in our technical report on benchmark text datasets [Rossi et al. 2013].

The experimental evaluation is based on a knowledge extraction task from text streams. The method of knowledge extraction is hierarchical clustering [Aggarwal and Reddy 2013], which is very popular in applications involving knowledge extraction from textual data, since it allows the organization of text documents at different levels of granularity (clusters and subclusters) and facilitates users to interactively explore and visualize the extracted knowledge [Zhao et al. 2005; Aggarwal and Zhai 2012].

In the experimental evaluation, the documents of the text database are presented sequentially to simulate a text stream. We conducted an unsupervised instance selection process and the selected instances are used to construct an initial hierarchical clustering. This initial model is used for incremental clustering considering the rest of the instances that were not selected in the process. Thus, if the unsupervised instance selection process is successful, then the initial hierarchical clustering model will present good performance to organize the rest of the database. In order to evaluate the performance of the knowledge extraction task, we use an evaluation criterion based on precision and recall, detailed in the next section.

4.1 Evaluation Criteria

The FScore index is a well-known measure that uses the ideas of precision and recall of the information retrieval field to evaluate the performance of a given model [Manning et al. 2008]. Larsen and Aone [1999] adapted the FScore index to evaluate models obtained by hierarchical clustering algorithms and, since then, the FScore index has been used in several studies to assess the quality of hierarchical organization of the knowledge extracted from texts [Zhao et al. 2005; Pang-Ning et al. 2006; Aggarwal and Reddy 2013]. This measure is used as an external criterion validation, because it uses prior knowledge (external information) about reference categories of the text database. The basic idea is to calculate how the hierarchical clustering was able to recover the category information associated with each instance of the database.

To calculate the FScore index, consider that

—H is a hierarchical clustering that represents the organization of the extracted knowledge;
—L_r is a category (external information) representing a set of instances of the same topic r; and
—G_i is a cluster, and its respective set of instances, belonging to hierarchical clustering H.

Thus, given a category L_r and a cluster G_i, we calculate the precision P and recall R measures according to Equation 1 and Equation 2 respectively. The harmonic mean F (Equation 3) is calculated to obtain a balance between precision and recall.

$$P(L_r, G_i) = \frac{|L_r \cap G_i|}{|G_i|} \qquad (1)$$

[1] National Science Foundation(USA): `http://www.nsf.gov/`

$$R(L_r, G_i) = \frac{|L_r \cap G_i|}{|L_r|} \quad (2)$$

$$F(L_r, G_i) = \frac{2 * P(L_r, G_i) * R(L_r, G_i)}{P(L_r, G_i) + R(L_r, G_i)} \quad (3)$$

The F measure selected for a given category L_r is the highest value obtained for a cluster belonging to hierarchy H, considering all the existing clusters and subclusters, according to Equation 4.

$$F(L_r) = \max_{G_i \in H} F(L_r, G_i) \quad (4)$$

Finally, the FScore value of the hierarchical clustering with n instances and c reference categories is calculated as the sum of F measures of each category weighted by the number of instances of the category (Equation 5).

$$FScore = \sum_{r=1}^{c} \frac{|L_r|}{n} F(L_r) \quad (5)$$

Thus, if the hierarchical clustering can recover the reference categories of the text database, then the FScore value is close to 1. Otherwise, FScore is close to 0.

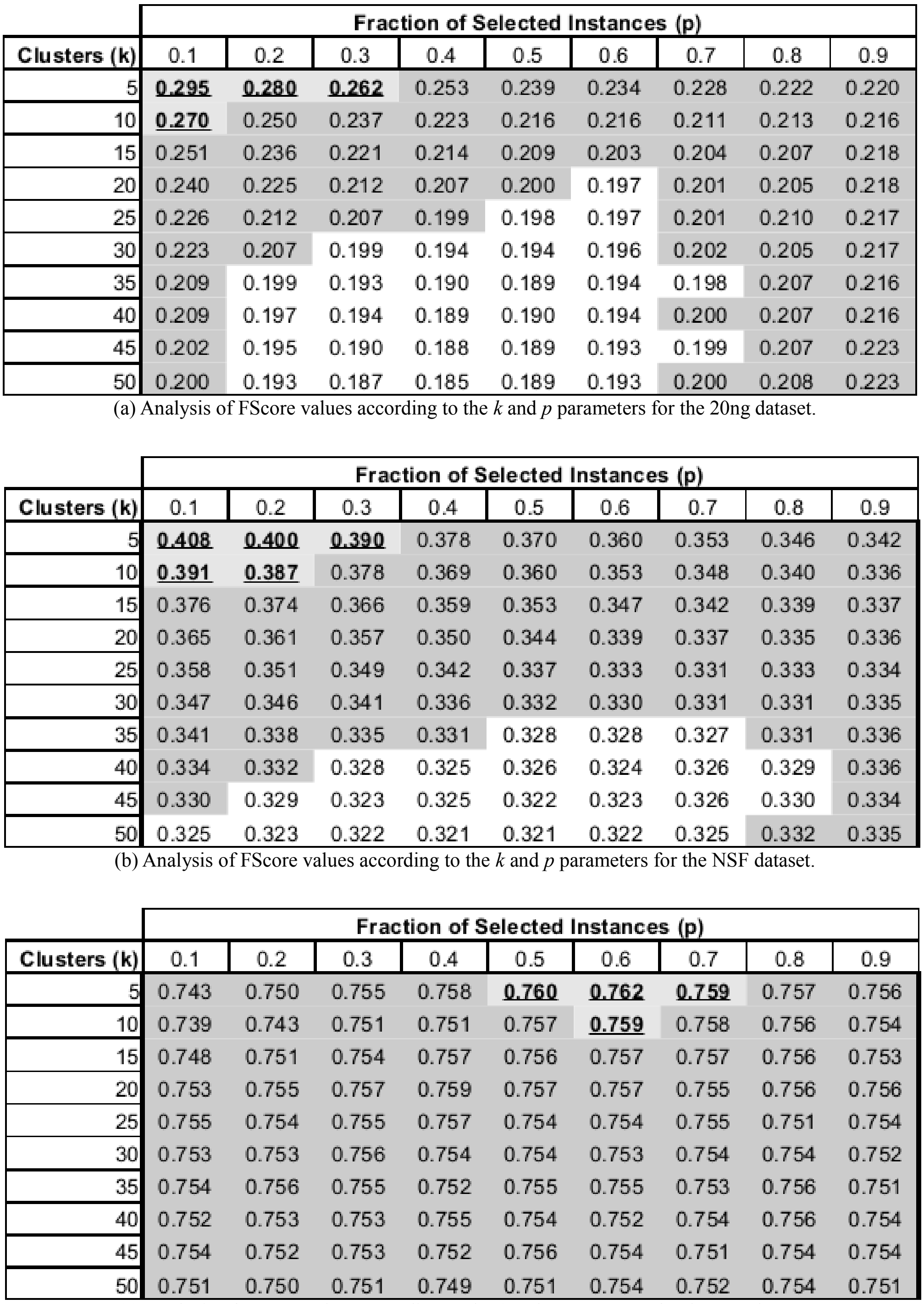
4.2 Experiment Setup

In the experiment setup, we used the the well-known partitioning clustering algorithm k-means and the cosine similarity measure to support the unsupervised instance selection. During the experiments, we analyzed the following numbers of clusters (k): $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. These ranges of values for k are commonly used in the cluster analysis literature [Milligan et al. 1983; Aggarwal and Reddy 2013], since it allows to analyze the behavior of the proposed approach considering several clustering granularities.

To identify the most representative instances, we apply the cosine similarity measure between the instances and the cluster centroids. Thus, we selected a fraction of the closest instances to the cluster centroids and analyzed the following values for the fraction of instances selected per cluster (p): 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; and 0.9.

The reservoir (and buffer) size m values were analyzed from 100 to 1000 for all text databases, thereby simulating scenarios with different memory requirements for the instance selection process. The evaluation process of the unsupervised instance selection was repeated 30 times in order to reduce the impact of random fluctuations in the result analysis and the average FScore values (and their standard deviation values) are presented.

The proposed approach in this article is identified as "Reservoir+Focusing" in the experimental evaluation, in reference to the incorporation of Focusing function described in Section 2. The results are compared with the traditional sampling technique of the *Reservoir Sampling*, identified as "Reservoir (traditional)". For statistical analysis, we used the non-parametric Wilcoxon signed-ranks test [Wilcoxon 1945], which is an alternative to the paired t-test. According to Demšar [2006], the Wilcoxon test is recommended for comparing performance between two algorithms and multiple datasets.



	Fraction of Selected Instances (p)								
Clusters (k)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	**0.295**	**0.280**	**0.262**	0.253	0.239	0.234	0.228	0.222	0.220
10	**0.270**	0.250	0.237	0.223	0.216	0.216	0.211	0.213	0.216
15	0.251	0.236	0.221	0.214	0.209	0.203	0.204	0.207	0.218
20	0.240	0.225	0.212	0.207	0.200	0.197	0.201	0.205	0.218
25	0.226	0.212	0.207	0.199	0.198	0.197	0.201	0.210	0.217
30	0.223	0.207	0.199	0.194	0.194	0.196	0.202	0.205	0.217
35	0.209	0.199	0.193	0.190	0.189	0.194	0.198	0.207	0.216
40	0.209	0.197	0.194	0.189	0.190	0.194	0.200	0.207	0.216
45	0.202	0.195	0.190	0.188	0.189	0.193	0.199	0.207	0.223
50	0.200	0.193	0.187	0.185	0.189	0.193	0.200	0.208	0.223

(a) Analysis of FScore values according to the *k* and *p* parameters for the 20ng dataset.

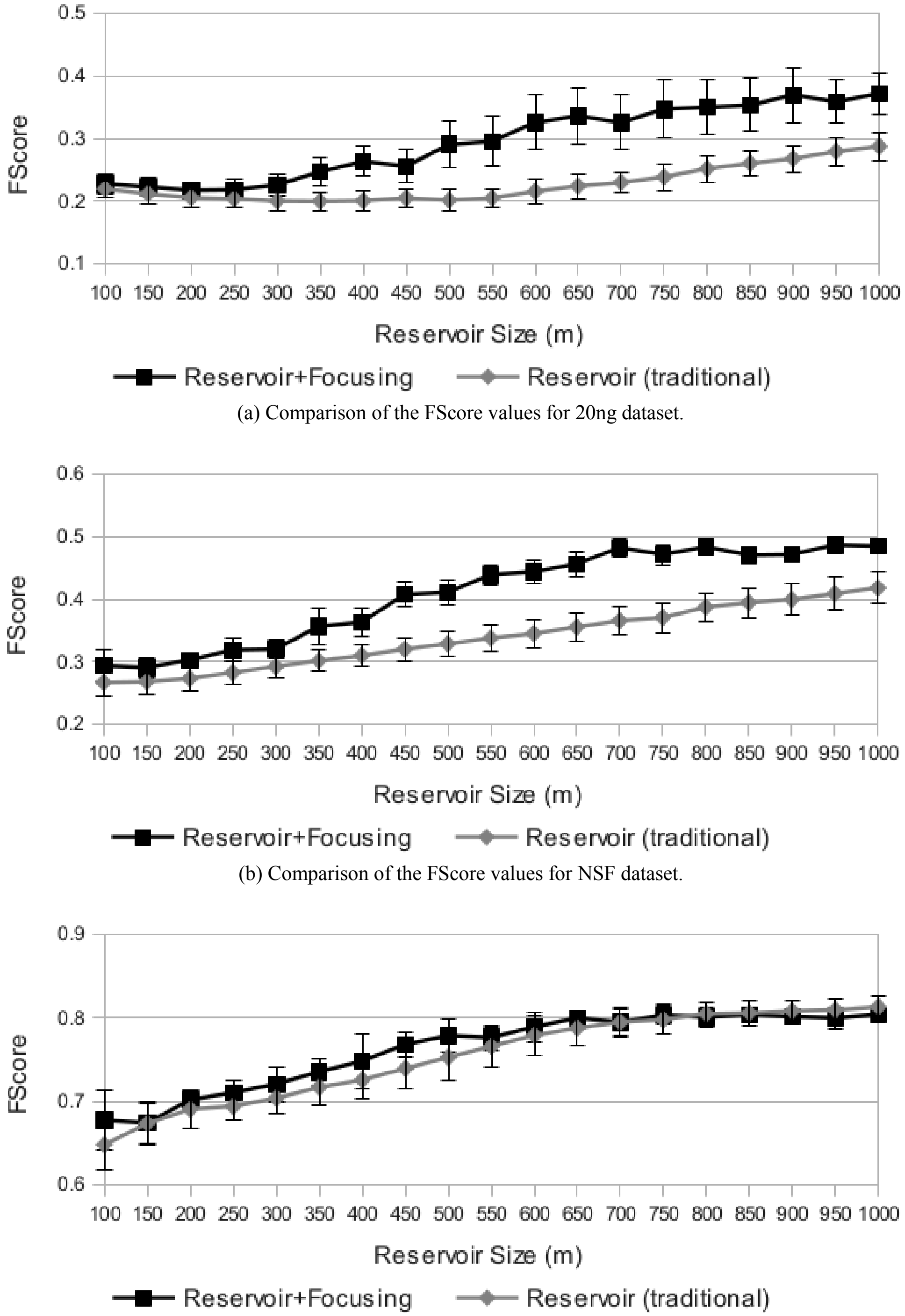
	Fraction of Selected Instances (p)								
Clusters (k)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	**0.408**	**0.400**	**0.390**	0.378	0.370	0.360	0.353	0.346	0.342
10	**0.391**	**0.387**	0.378	0.369	0.360	0.353	0.348	0.340	0.336
15	0.376	0.374	0.366	0.359	0.353	0.347	0.342	0.339	0.337
20	0.365	0.361	0.357	0.350	0.344	0.339	0.337	0.335	0.336
25	0.358	0.351	0.349	0.342	0.337	0.333	0.331	0.333	0.334
30	0.347	0.346	0.341	0.336	0.332	0.330	0.331	0.331	0.335
35	0.341	0.338	0.335	0.331	0.328	0.328	0.327	0.331	0.336
40	0.334	0.332	0.328	0.325	0.326	0.324	0.326	0.329	0.336
45	0.330	0.329	0.323	0.325	0.322	0.323	0.326	0.330	0.334
50	0.325	0.323	0.322	0.321	0.321	0.322	0.325	0.332	0.335

(b) Analysis of FScore values according to the *k* and *p* parameters for the NSF dataset.

	Fraction of Selected Instances (p)								
Clusters (k)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	0.743	0.750	0.755	0.758	**0.760**	**0.762**	**0.759**	0.757	0.756
10	0.739	0.743	0.751	0.751	0.757	**0.759**	0.758	0.756	0.754
15	0.748	0.751	0.754	0.757	0.756	0.757	0.757	0.756	0.753
20	0.753	0.755	0.757	0.759	0.757	0.757	0.755	0.756	0.756
25	0.755	0.754	0.755	0.757	0.754	0.754	0.755	0.751	0.754
30	0.753	0.753	0.756	0.754	0.754	0.753	0.754	0.754	0.752
35	0.754	0.756	0.755	0.752	0.755	0.755	0.753	0.756	0.751
40	0.752	0.753	0.753	0.755	0.754	0.752	0.754	0.756	0.754
45	0.754	0.752	0.753	0.752	0.756	0.754	0.751	0.754	0.754
50	0.751	0.750	0.751	0.749	0.751	0.754	0.752	0.754	0.751

(c) Analysis of FScore values according to the *k* and *p* parameters for the Re8 dataset.

Fig. 1. Analysis of parameters k (number of clusters) and p (fraction of selected instances per cluster) of the proposed approach. FScore values shown in bold and underlined represent settings where the proposed approach ("Reservoir+Focusing") obtains statistically superior results compared to traditional Reservoir. Gray cells indicate the settings in which there is no statistically significant difference. White cells indicate the settings where traditional Reservoir achieves statistically superior results.



(a) Comparison of the FScore values for 20ng dataset.

(b) Comparison of the FScore values for NSF dataset.

(c) Comparison of the FScore values for Re8 dataset.

Fig. 2. FScore values of the models (hierarchical clustering) constructed from the instances selected by the proposed approach (Reservoir+Focusing) compared with traditional Reservoir.

4.3 Results and Discussion

The experimental results are discussed considering two aspects: (1) analysis of the parameters k (number of clusters) and p (fraction of selected instances per cluster) of the proposed approach, and (2) comparison of the unsupervised instance selection between "Reservoir+Focusing" and "Reservoir (Traditional)", according to the size (m) of the reservoir.

Figure 1 summarizes the statistical analysis of parameters of the proposed approach. For each dataset, FScore values are presented in a table with the k and p values. When a particular setting of the proposed approach obtains (statistically) superior results compared with the "Reservoir (traditional)", then the values are shown in bold and underlined. When there is no statistical difference, then the table cell is colored with gray. Finally, when a particular setting of the proposed approach obtains inferior results than the "Reservoir (traditional)", then the cell is colored with white.

According to the results, for the analyzed text databases, the unsupervised instance selection is improved by the use of a small number of clusters and also selecting few instances of each cluster, i.e., a low value for p. In this case, only the instances located in the center of high-density regions are used in the *Reservoir Sampling*. This is a promising result, since the use of a small number of clusters is less computationally expensive, and obtains superior results than the "Reservoir (traditional)".

Figure 2 presents the comparison of the unsupervised instance selection between "Reservoir+Focusing" and "Reservoir (Traditional)". The results are presented considering various values of the reservoir size (m) and the best settings of the parameters k and p for each text database are considered. The parameter values used were: $k = 5$ and $p = 0.1$ for the 20ng dataset; $k = 5$ and $p = 0.1$ for the NSF dataset; and $k = 5$ and $p = 0.6$ for the Re8 dataset.

In general, we have observed that the unsupervised instance selection improves as the size of the reservoir increases, since there are more instances to identify the separation boundaries. Our approach improves the performance of the hierarchical clustering model, particularly for databases that present class overlapping (complex problems), which is the case of the 20ng and NSF text databases. If the database contains low class overlapping (such as Re8 database), our approach presents an improvement in the hierarchical clustering only when we use a small reservoir size. In this case, an instance selection based only on data sampling may provide competitive results. However, it is important to emphasize that, in real scenarios, there is no external information about the classes and complexity of the problems, especially in unsupervised settings.

5. CONCLUDING REMARKS

In this article, we presented an approach for unsupervised instance selection from text streams. The reported experimental results demonstrate that the proposed approach increases the performance of the knowledge extraction task based on hierarchical clustering, since it uses more representative instances to build an initial model clustering. Although the experimental evaluation presented in this work is based on hierarchical clustering, the proposed approach can be used to select representative instances for various knowledge extraction process. More details of the experimental evaluation, as well as the algorithms used in this work are available online at `http://sites.labic.icmc.usp.br/torch/jidm2013`.

The proposed approach in this article is potentially promising for various tasks involving text streams. Identifying a relevant and representative subset of instances is useful for data and text mining and, more recently, for many applications related to *big data*. An advantage of the proposed approach is to allow the reuse of several existing algorithms for knowledge extraction and machine learning – by reducing the size of the database to be analyzed. Moreover, our approach uses a simple strategy and is easily adaptable to other data types (non textual data) by just changing the clustering method and similarity measure used for instance selection.

Directions for future work involve the use of semi-supervised clustering methods to support the instance selection. Unlike supervised methods, which require a large set of labeled data, semi-supervised methods require only a small set of constraints or user's feedback. Thus, it is possible to guide the instance selection process according to the user's expectations and existing background knowledge about the problem domain.

REFERENCES

Aggarwal, C. C. Mining Text Streams. In C. C. Aggarwal and C. Zhai (Eds.), *Mining Text Data*. Springer, New York, pp. 297–321, 2012.

Aggarwal, C. C. and Reddy, C. K. *Data Clustering: Algorithms and Applications*. CRC Press, New York, 2013.

Aggarwal, C. C. and Zhai, C. A Survey of Text Clustering Algorithms. In C. C. Aggarwal and C. Zhai (Eds.), *Mining Text Data*. Springer, New York, pp. 77–128, 2012.

Czarnowski, I. Cluster-Based Instance Selection for Machine Classification. *Knowledge and Information Systems* 30 (1): 113–133, 2012.

Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* vol. 7, pp. 1–30, 2006.

Gama, J. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 2010.

Gama, J. A Survey on Learning from Data Streams: current and future trends. *Progress in Artificial Intelligence* 1 (1): 45–55, 2012.

Larsen, B. and Aone, C. Fast and effective text mining using linear-time document clustering. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. pp. 16–22, 1999.

Lewis, D. D. Reuters-21578 Text Categorization Test Collection. Accessed: 2014-03-10, 1997. Available at http://www.daviddlewis.com/resources/testcollections/.

Leyva, E., Gonzalez, A., and Perez, R. Knowledge-Based Instance Selection: a compromise between efficiency and versatility. *Knowledge-Based Systems* 47 (0): 65 – 76, 2013.

Liu, H. *Instance Selection and Construction for Data Mining*. Springer-Verlag, 2010.

Liu, H. and Motoda, H. On Issues of Instance Selection. *Data Mining and Knowledge Discovery* 6 (2): 115–130, 2002.

MacQueen, J. B. Some Methods for Classification and Analysis of MultiVariate Observations. In *Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. pp. 281–297, 1967.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge Press, 2008.

McLeod, A. I. and Bellhouse, D. R. A Convenient Algorithm for Drawing a Simple Random Sample. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 32 (2): 182–184, 1983.

Milligan, G. W., Soon, S. C., and Sokol, L. M. The Effect of Cluster Size, Dimensionality, and the Number of Clusters on Recovery of True Cluster Structure. *Pattern Analysis and Machine Intelligence* 5 (1): 40–47, 1983.

Monahan, J. F. *Numerical Methods of Statistics*. Cambridge University Press, 2011.

Olvera-Lopez, J. A., Carrasco-Ochoa, J. A., Martidnez-Trinidad, J. F., and Kittler, J. A Review of Instance Selection Methods. *Artificial Intelligence Review* 34 (2): 133–143, 2010.

Pang, S. CSMINING Group - The R8 of Reuters 21578 Data Set. Accessed: 2014-03-10, 2010. Available at http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html.

Pang-Ning, T., Steinbach, M., Kumar, V., et al. *Introduction to Data Mining*. Addison-Wesley Companion, 2006.

Pazzani, M. J. and Meyers, A. NSF Research Award Abstracts 1990-2003 Data Set. Accessed: 2014-03-10, 2003. Available at http://archive.ics.uci.edu/ml/databases/nsfabs/.

Reinartz, T. A Unifying View on Instance Selection. *Data Mining and Knowledge Discovery* 6 (2): 191–210, 2002.

Rennie, J. The 20 Newsgroups Dataset. MIT Computer Science and Artificial Intelligence Laboratory. Accessed: 2014-03-10, 2008. Available at http://people.csail.mit.edu/jrennie/20Newsgroups/.

Rossi, R. G., Marcacini, R. M., and Rezende, S. O. Benchmarking Text Collections for Classification and Clustering Tasks. Technical Report. Institute of Mathematics and Computer Sciences - University of São Paulo, 2013. http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_RT_395.pdf.

Vitter, J. S. Random Sampling with a Reservoir. *Transactions on Mathematical Software* 11 (1): 37–57, 1985.

Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biometrics* 1 (6): 80–83, 1945.

Zhao, Y., Karypis, G., and Fayyad, U. Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery* 10 (2): 141–168, 2005.