

Is Unlabeled Data Suitable for Multiclass SVM-based Web Page Classification?

Arkaitz Zubiaga, Víctor Fresno, Raquel Martínez

NLP & IR Group at UNED

Lenguajes y Sistemas Informáticos

E.T.S.I. Informática, UNED

{azubiaga, vfresno, raquel}@lsi.uned.es

Abstract

Support Vector Machines present an interesting and effective approach to solve automated classification tasks. Although it only handles binary and supervised problems by nature, it has been transformed into multiclass and semi-supervised approaches in several works. A previous study on supervised and semi-supervised SVM classification over binary taxonomies showed how the latter clearly outperforms the former, proving the suitability of unlabeled data for the learning phase in this kind of tasks. However, the suitability of unlabeled data for multiclass tasks using SVM has never been tested before. In this work, we present a study on whether unlabeled data could improve results for multiclass web page classification tasks using Support Vector Machines. As a conclusion, we encourage to rely only on labeled data, both for improving (or at least equaling) performance and for reducing the computational cost.

1 Introduction

The amount of web documents is increasing in a very fast way in the last years, what makes more and more complicated its organization. For this reason, web page classification has gained importance as a task to ease and improve information access. Web page classification can be defined as the task of labeling and organizing web documents within a set of predefined categories. In this work, we focus on web page classification based on Support Vector Machines (SVM, (Joachims, 1998)). This kind of classification tasks rely on a previously labeled

training set of documents, with which the classifier acquires the required ability to classify new unknown documents.

Different settings can be distinguished for web page classification problems. On the one hand, attending to the learning technique the system bases on, it may be *supervised*, with all the training documents previously labeled, or *semi-supervised*, where unlabeled documents are also taken into account during the learning phase. On the other hand, attending to the number of classes, the classification may be *binary*, where only two possible categories can be assigned to each document, or *multiclass*, where three or more categories can be set. The former is commonly used for filtering systems, whereas the latter is necessary for bigger taxonomies, e.g. topical classification.

Although multiple studies have been made for text classification, its application to the web page classification area remains without enough attention (Qi and Davison, 2007). Analyzing the nature of a web page classification task, we can consider it to be, generally, multiclass problems, where it is usual to find numerous classes. In the same way, if we take into account that the number of available labeled documents is tiny compared to the size of the Web, this task becomes semi-supervised besides multiclass.

However, the original SVM algorithm supports neither semi-supervised learning nor multiclass taxonomies, due to its dichotomic and supervised nature. To solve this issue, different studies for both multiclass SVM and semi-supervised SVM approaches have been proposed, but a little effort has

been invested in the combination of them.

(Joachims, 1999) compares supervised and semi-supervised approaches for binary tasks using SVM. It shows encouraging results for the transductive semi-supervised approach, clearly improving the supervised, and so he proved unlabeled data to be suitable to optimize binary SVM classifiers' results. On the other hand, the few works presented for semi-supervised multiclass SVM classification do not provide clear information on whether the unlabeled data improves the classification results in comparison with the only use of labeled data.

In this work, we performed an experiment among different SVM-based multiclass approaches, both supervised and semi-supervised. The experiments were focused on web page classification, and were carried out over three benchmark datasets: *BankSearch*, *WebKB* and *Yahoo! Science*. Using the results of the comparison, we analyze and study the suitability of unlabeled data for multiclass SVM classification tasks. We discuss these results and evaluate whether it is worthy to rely on a semi-supervised SVM approach to conduct this kind of tasks.

The remainder of this document is organized as follows. Next, in section 2, we briefly explain how SVM classifiers work for binary classifications, both for a supervised and a semi-supervised view. In section 3, we continue with the adaptation of SVM to multiclass environments, and show what has been done in the literature. Section 4 presents the details of the experiments carried out in this work, aim at evaluating the suitability of unlabeled data for multiclass SVM classification. In section 5 we show and discuss the results of the experiments. Finally, in section 6, we conclude with our thoughts and future work.

2 Binary SVM

In the last decade, SVM has become one of the most studied techniques for text classification, due to the positive results it has shown. This technique uses the vector space model for the documents' representation, and assumes that documents in the same class should fall into separable spaces of the representation. Upon this, it looks for a hyperplane that separates the classes; therefore, this hyperplane should

maximize the distance between it and the nearest documents, what is called the margin. The following function is used to define the hyperplane (see Figure 1):

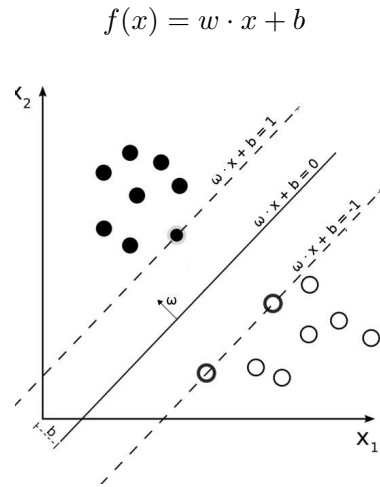


Figure 1: An example of binary SVM classification, separating two classes (black dots from white dots)

In order to resolve this function, all the possible values should be considered and, after that, the values of w and b that maximize the margin should be selected. This would be computationally expensive, so the following equivalent function is used to relax it (Boser et al. , 1992) (Cortes and Vapnik, 1995):

$$\min \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i^d \right]$$

$$\text{Subject to: } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

where C is the penalty parameter, ξ_i is an slack variable for the i^{th} document, and l is the number of labeled documents.

This function can only resolve linearly separable problems, thus the use of a kernel function is commonly required for the redimension of the space; in this manner, the new space will be linearly separable. After that, the redimension is undone, so the found hyperplane will be transformed to the original space, respecting the classification function. Best-known kernel functions include linear, polynomial, radial basis function (RBF) and sigmoid, among others. Different kernel functions' performance has been studied in (Schölkopf and Smola, 1999) and (Kivinen et al., 2002).

Note that the function above can only resolve binary and supervised problems, so different variants are necessary to perform semi-supervised or multi-class tasks.

2.1 Semi-supervised Learning for SVM (S³VM)

Semi-supervised learning approaches differ in the learning phase. As opposed to supervised approaches, unlabeled data is used during the learning phase, and so classifier’s predictions over them is also included as labeled data to learn. The fact of taking into account unlabeled data to learn can improve the classification done by supervised methods, specially when its predictions provide new useful information, as shown in figure 2. However, the noise added by erroneous predictions can make worse the learning phase and, therefore, its final performance. This makes interesting the study on whether relying on semi-supervised approaches is suitable for each kind of task.

Semi-supervised learning for SVM, also known as S³VM, was first introduced by (Joachims, 1999) in a transductive way, by modifying the original SVM function. To do that, he proposed to add an additional term to the optimization function:

$$\min \left[\frac{1}{2} \cdot \|\omega\|^2 + C \cdot \sum_{i=1}^l \xi_i^d + C^* \cdot \sum_{j=1}^u \xi_j^{*d} \right]$$

where u is the number of unlabeled data.

Nevertheless, the adaptation of SVM to semi-supervised learning significantly increases its computational cost, due to the non-convex nature of the resulting function, and so obtaining the minimum value is even more complicated. In order to relax the function, convex optimization techniques such as semi-definite programming are commonly used (Xu et al. , 2007), where minimizing the function gets much easier.

By means of this approach, (Joachims, 1999) demonstrated a large performance gap between the original supervised SVM and his semi-supervised proposal, in favour of the latter one. He showed that for binary classification tasks, the smaller is the training set size, the larger gets the difference among these two approaches. Although he worked

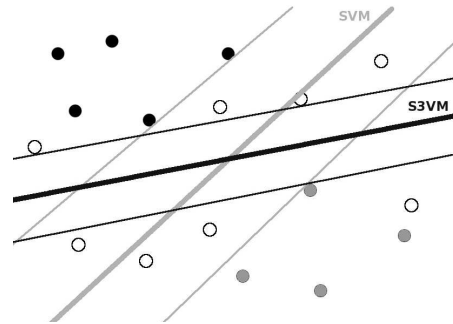


Figure 2: SVM vs S³VM, where white balls are unlabeled documents

with multiclass datasets, he splitted the problems into smaller binary ones, and so he did not demonstrate whether the same performance gap occurs for multiclass classification. This paper tries to cover this issue. (Chapelle et al., 2008) present a comprehensive study on S³VM approaches.

3 Multiclass SVM

Due to the dichotomic nature of SVM, it came up the need to implement new methods to solve multiclass problems, where more than two classes must be considered. Different approaches have been proposed to achieve this. On the one hand, as a direct approach, (Weston, 1999) proposed modifying the optimization function getting into account all the k classes at once:

$$\min \left[\frac{1}{2} \sum_{m=1}^k \|w_m\|^2 + C \sum_{i=1}^l \sum_{m \neq y_i} \xi_i^m \right]$$

Subject to:

$$w_{y_i} \cdot x_i + b_{y_i} \geq w_m \cdot x_i + b_m + 2 - \xi_i^m, \xi_i^m \geq 0$$

On the other hand, the original binary SVM classifier has usually been combined to obtain a multiclass solution. As combinations of binary SVM classifiers, two different approaches to k -class classifiers can be emphasized (Hsu and Lin, 2002):

- *one-against-all* constructs k classifiers defining that many hyperplanes; each of them separates the class i from the rest $k-1$. For instance, for a problem with 4 classes, 1 vs $2-3-4$, 2 vs $1-3-4$, 3 vs $1-2-4$ and 4 vs $1-2-3$ classifiers would

be created. New documents will be categorized in the class of the classifier that maximizes the margin: $\hat{C}_i = \arg \max_{i=1, \dots, k} (w_i x + b_i)$. As the number of classes increases, the amount of classifiers will increase linearly.

- *one-against-one* constructs $\frac{k(k-1)}{2}$ classifiers, one for each possible category pair. For instance, for a problem with 4 classes, *1 vs 2*, *1 vs 3*, *1 vs 4*, *2 vs 3*, *2 vs 4* and *3 vs 4* classifiers would be created. After that, it classifies each new document by using all the classifiers, where a vote is added for the winning class over each classifier; the method will propose the class with more votes as the result. As the number of classes increases, the amount of classifiers will increase in an exponential way, and so the problem could become very expensive for large taxonomies.

Both (Weston, 1999) and (Hsu and Lin, 2002) compare the direct multiclass approach to the *one-against-one* and *one-against-all* binary classifier combining approaches. They agree concluding that the direct approach does not outperform the results by *one-against-one* nor *one-against-all*, although it considerably reduces the computational cost because the number of support vector machines it constructs is lower. Among the binary combining approaches, they show the performance of *one-against-one* to be superior to *one-against-all*.

Although these approaches have been widely used in supervised learning environments, they have scarcely been applied to semi-supervised learning. Because of this, we believe the study on its applicability and performance for this type of problems could be interesting.

3.1 Multiclass S³VM

When the taxonomy is defined by more than two classes and the number of previously labeled documents is very small, the combination of both multiclass and semi-supervised approaches could be required. That is, a multiclass S³VM approach. The usual web page classification problem meets with these characteristics, since more than two classes are usually needed, and the tiny amount of labeled documents requires the use of unlabeled data for the learning phase.

Actually, there are a few works focused on transforming SVM into a semi-supervised and multiclass approach. As a direct approach, a proposal by (Yajima and Kuo, 2006) can be found. They modify the function for multiclass SVM classification and get it usable for semi-supervised tasks. The resulting optimization function is as follows:

$$\min \frac{1}{2} \sum_{i=1}^h \beta^{i^T} K^{-1} \beta^i + C \sum_{j=1}^l \sum_{i \neq y_j} \max\{0, 1 - (\beta_j^{y_j} - \beta_j^i)\}^2$$

where β represents the product of a vector of variables and a kernel matrix defined by the author.

On the other hand, some other works are based on different approaches to achieve a multiclass S³VM classifier.

(Qi et al., 2004) use Fuzzy C-Means (FCM) to predict labels for unlabeled documents. After that, multiclass SVM is used to learn with the augmented training set, classifying the test set. (Xu y Schuurmans, 2005) rely on a clustering-based approach to label the unlabeled data. Afterwards, they apply a multiclass SVM classifier to the fully labeled training set. (Chapelle et al., 2006) present a direct multiclass S³VM approach by using the Continuation Method. On the other hand, this is the only work, to the best of our knowledge, that has tested the *one-against-all* and *one-against-one* approaches in a semi-supervised environment. They apply these methods to some news datasets, for which they get low performance. Additionally, they show that *one-against-one* is not sufficient for real-world multiclass semi-supervised learning, since the unlabeled data cannot be restricted to the two classes under consideration.

It is noteworthy that most of the above works only presented their approaches and compared them to other semi-supervised classifying methods, such as Expectation-Maximization (EM) or Naive Bayes. As an exception, (Chapelle et al., 2006) compared a semi-supervised and a supervised SVM approach, but only over image datasets. Against this, we felt the need to evaluate and compare multiclass SVM and multiclass S³VM approaches, for the sake of discovering whether learning with unlabeled web

documents is helpful for multiclass problems when using SVM as a classifier.

4 Multiclass SVM versus Multiclass S^3VM

The main goal of this work is to evaluate the real contribution of unlabeled data for multiclass SVM-based web page classification tasks. There are a few works using semi-supervised multiclass SVM classifiers, but nobody has demonstrated it improves supervised SVM classifier's performance. Next, we detail the experiments we carried out to clear up any doubts and to ensure which is better for multiclass SVM-based web page classifications.

4.1 Approaches

In order to evaluate and compare multiclass SVM and multiclass S^3VM , we decided to use three different but equivalent approaches for each view, supervised and semi-supervised. For further information on these approaches, see section 3. We add a suffix, *-SVM* or *- S^3VM* , to the names of the approaches, to differentiate whether they are based in a supervised or a semi-supervised algorithm.

On the part of the semi-supervised view, the following three approaches were selected:

- *2-steps-SVM*: we called *2-steps-SVM* to the technique based on the direct multiclass supervised approach exposed in section 3. This method works, on its first step, with the training collection, learning with the labeled documents and predicting the unlabeled ones; after that, the latter documents are labeled based on the generated predictions. On the second step, now with a fully labeled training set, the usual supervised classification process is done, learning with the training documents and predicting the documents in the test set.

This approach is somehow similar to those proposed by (Qi et al., 2004) and (Xu y Schuurmans, 2005). Nonetheless, the *2-steps-SVM* approach uses the same method for both the first and second steps. A supervised multiclass SVM is used to increase the labeled set and, after that, to classify the test set.

- *one-against-all- S^3VM* : the *one-against-all* approach has not sufficiently been tested for semi-

supervised environments, and seems interesting to evaluate its performance.

- *one-against-one- S^3VM* : the *one-against-one* does not seem to be suitable for semi-supervised environments, since the classifier is not able to ignore the inadequate unlabeled documents for each 1-vs-1 binary task, as stated by (Chapelle et al., 2006). Anyway, since it has scarcely been tested, we also consider this approach.

On the other hand, the approaches selected for the supervised view were these: (1) *1-step-SVM*; (2) *one-against-all-SVM*, and (3) *one-against-one-SVM*.

The three approaches mentioned above are analogous to the semi-supervised approaches, *2-steps-SVM*, *one-against-all- S^3VM* and *one-against-one- S^3VM* , respectively. They differ in the learning phase: unlike the semi-supervised approaches, these three supervised approaches only rely on the labeled documents for the learning task, but after that they classify the same test documents. These approaches allow to evaluate whether the unlabeled documents are contributing in a positive or negative way in the learning phase.

4.2 Datasets

For these experiments we have used three web page benchmark datasets previously used for classification tasks:

- *BankSearch* (Sinka and Corne, 2002), a collection of 11,000 web pages over 11 classes, with very different topics: commercial banks, building societies, insurance agencies, java, c, visual basic, astronomy, biology, soccer, motorsports and sports. We removed the category sports, since it includes both soccer and motorsports in it, as a parent category. This results 10,000 web pages over 10 categories. 4,000 instances were assigned to the training set, while the other 6,000 were left on the test set.
- *WebKB*¹, with a total of 4,518 documents of 4 universities, and classified into 7 classes

¹<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

(student, faculty, personal, department, course, project and other). The class named *other* was removed due to its ambiguity, and so we finally got 6 classes. 2,000 instances fell into the training set, and 2,518 to the test set.

- *Yahoo! Science* (Tan, 2002), with 788 scientific documents, classified into 6 classes (agriculture, biology, earth science, math, chemistry and others). We selected 200 documents for the training set, and 588 for the test set.

Within the training set, for each dataset, multiple versions were created, modifying the number of labeled documents, while the rest were left unlabeled. Thus, the size of labeled subset within the training set changes, ranging from 50 web documents to the whole training set.

4.3 Document Representation

SVM requires a vectorial representation of the documents as an input for the classifier, both for train and test phases. To obtain this vectorial representation, we first converted the original html files into plain text files, removing all the html tags. After that, we removed the noisy tokens, such as URLs, email addresses or some stopwords. For these edited documents, the *tf-idf* term weighting function was used to define the values for the uniterms found on the texts. As the term dimensionality became too large, we then removed the least-frequent terms by its document frequency; terms appearing in less than 0.5% of the documents were removed for the representation. The remaining uniterms define the vector space dimensions. That derived term vectors with 8285 dimensions for *BankSearch* dataset, 3115 for *WebKB* and 8437 for *Yahoo! Science*.

4.4 Implementation

To carry out our experiments, we based on freely available and already tested and experimented software. Different SVM classifiers were needed to implement the methods described in section 4.1.

SVMLight² was used to work with binary semi-supervised classifiers for the *one-against-all-S³VM* and *one-against-one-S³VM* approaches. In the same way, we implemented their supervised versions,

²<http://svmlight.joachims.org>

one-against-all-SVM and *one-against-one-SVM*, in order to evaluate the contribution of unlabeled data. To achieve the supervised approaches, we ignored the unlabeled data during the training phase and, after that, tested with the same test set used for semi-supervised approaches. The default settings using a polynomial kernel were selected for the experiments.

SVMmulticlass³ was used to implement the *2-steps-SVM* approach, by using it two times. Firstly, to train the labeled data and classify unlabeled data. After that, to train with the whole training set labeled with classifier's predictions, and to test with the test set. In the same way, the *1-step-SVM* method was implemented by ignoring unlabeled data and training only the labeled data. This method allows to evaluate the contribution of unlabeled data for the *2-steps-SVM* method.

4.5 Evaluation Measures

For the evaluation of the experiments we used the *accuracy* to measure the performance, since it has been frequently used for text classification problems, specially for multiclass tasks. The accuracy offers the percent of the correct predictions for the whole test set. We have considered the same weight for all the correct guesses for any class. A correct prediction in any of the classes has the same value, thus no weighting exists.

On the other hand, an averaged accuracy evaluation is also possible for the binary combining approaches. An averaged accuracy makes possible to evaluate the results by each binary classifier, and provides an averaged value for the whole binary classifier set. It is worth to note that these values do not provide any information for the evaluation of the combined multiclass results, but only for evaluating each binary classifier before combining them.

5 Results and Discussion

Next, we show and discuss the results of our experiments. It is remarkable that both *one-against-one-SVM* and *one-against-one-S³VM* approaches were very inferior to the rest, and so we decided not to plot them in order to maintain graphs' clarity. Hence, figures 3, 4 and 5 show the results in accordance

³http://www.cs.cornell.edu/People/tj/svm_light/svm_multiclass.html

with the labeled subset size for the *2-steps-SVM*, *1-step-SVM*, *one-against-all-S³VM* and *one-against-all-SVM* approaches within our experiments. For the results to be more representative, nine executions were done for each subset, obtaining the mean value. These nine executions vary on the labeled subset within the training set.

The fact that *one-against-one-S³VM* has been the worst approach for our experiments confirms that the noise added by the unlabeled documents within each 1-vs-1 binary classification task is harmful to the learning phase, and it is not corrected when merging all the binary tasks.

The averaged accuracy for the combined binary classifiers allows to compare the *one-against-one* and *one-against-all* views. The averaged accuracy for *one-against-one-S³VM* shows very low performance (about 60% in most cases), whereas the same value for *one-against-all-S³VM* is much higher (about 90% in most cases). This is obvious to happen for the *one-against-all* view, since it is much easier to predict documents not pertaining to the class under consideration for each *1-vs-all* binary classifier. Although each binary classifier gets about 90% accuracy for the *one-against-one-S³VM* approach, this value falls considerably when combining them to get the multiclass result. This shows the additional difficulty for multiclass problems compared to binary ones. Hence, the difficulty to correctly predict unlabeled data increases for multiclass tasks, and it is more likely to add noise during the learning phase.

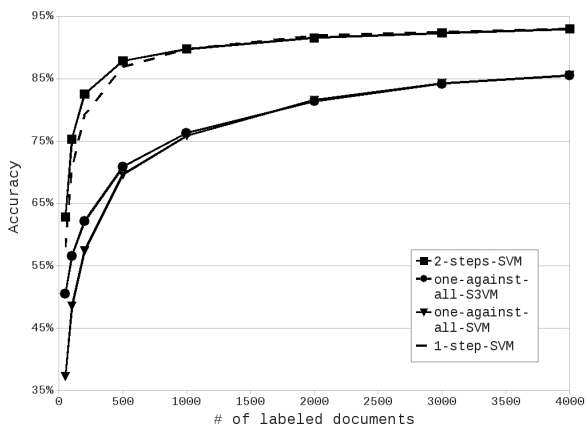


Figure 3: Results for BankSearch dataset

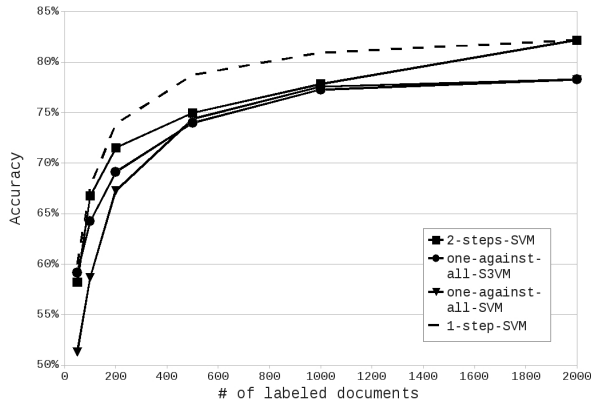


Figure 4: Results for WebKB dataset

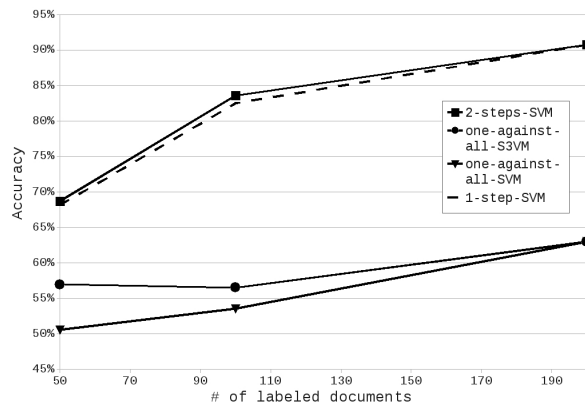


Figure 5: Results for Yahoo! Science dataset

For all the datasets we worked with, there is a noticeable performance gap between direct multiclass and binary combining approaches. Both *2-steps-SVM* and *1-step-SVM* are always on the top of the graphs, and *one-against-all-S³VM* and *one-against-all-SVM* approaches are so far from catching up with their results, except for *WebKB* dataset, where the gap is not so noticeable. This seems encouraging, since considering less support vectors in a direct multiclass approach reduces the computational cost and improves the final results.

Comparing the two analogous approaches among them, different conclusions could be extracted. On the one hand, *one-against-all-S³VM* shows slightly better results than *one-against-all-SVM*, and so considering unlabeled documents seems to be

favourable for the *one-against-all* view. On the other hand, the direct multiclass view shows varying results. Both *2-steps-SVM* and *1-step-SVM* show very similar results for *BankSearch* and *Yahoo! Science* datasets, but superior for *1-step-SVM* over the *WebKB* dataset. As a conclusion of this, ignoring unlabeled documents by means of the *1-step-SVM* approach seems to be advisable, since it reduces the computation cost, obtaining at least the same results than the semi-supervised *2-steps-SVM*.

Although their results are so poor, as we said above, the supervised approach wins for the *one-against-one* view; this confirms, again, that the *one-against-one* view is not an adequate view to be applied in a semi-supervised environment, due to the noise existing during the learning phase.

When analyzing the performance gaps between the analogous approaches, a general conclusion can be extracted: the smaller is the labeled subset the bigger is the performance gap, except for the *Yahoo! Science* dataset. Comparing the two best approaches, *1-step-SVM* and *2-steps-SVM*, the performance gap increases when the number of labeled documents decrease for *BankSearch*; for this dataset, the accuracy by *1-step-SVM* is 0.92 times the one by *2-steps-SVM* when the number of labeled documents is only 50, but this proportion goes to 0.99 with 500 labeled documents. This reflects how the contribution of unlabeled data decreases while the labeled set increases. For *WebKB*, the performance gap is in favour of *1-step-SVM*, and varies between 1.01 and 1.05 times *2-steps-SVM* method's accuracy, even with only 50 labeled documents. Again, increasing the labeled set negatively affects semi-supervised algorithm's performance. Last, for *Yahoo! Science*, the performance gap among these two approaches is not considerable, since their results are very similar.

Our conjecture for the performance difference between *1-step-SVM* and *2-steps-SVM* for the three datasets is the nature of the classes. The accuracy by semi-supervised *2-steps-SVM* is slightly higher for *BankSearch* and *Yahoo! Science*, where the classes are quite heterogeneous. On the other hand, the accuracy by supervised *1-step-SVM* is clearly higher for *WebKB*, where all the classes are an academic topic, and so more homogeneous. The semi-supervised classifiers show a major problem for pre-

dicting the unlabeled documents when the collection is more homogeneous, and so more difficult to differ between classes.

In summary, the main idea is that unlabeled documents do not seem to contribute as they would for multiclass tasks using SVM. Within the approaches we tested, the supervised *1-step-SVM* approach shows the best (or very similar to the best in some cases) results in accuracy and, taking into account it is the least-expensive approach, we strongly encourage to use this approach to solve multiclass web page classification tasks, mainly when the classes under consideration are homogeneous.

6 Conclusions and Outlook

We have studied and analyzed the contribution of considering unlabeled data during the learning phase for multiclass web page classification tasks using SVM. Our results show that ignoring unlabeled document to learn reduces computational cost and, additionally, obtains similar or slightly worse accuracy values for heterogeneous taxonomies, but higher for homogeneous ones. Therefore we show that, unlike for binary cases, as was shown by (Joachims, 1999), a supervised view outperforms a semi-supervised one for multiclass environments. Our thought is that predicting unlabeled documents' class is much more difficult when the number of classes increases, and so, the mistaken labeled documents are harmful for classifier's learning phase.

As a future work, a direct semi-supervised multiclass approach, such as those proposed by (Yajima and Kuo, 2006) and (Chapelle et al., 2006), should also be considered, as well as setting the classifier with different parameters or kernels. Balancing the weight of previously and newly labeled data could also be interesting to improve semi-supervised approaches' results.

Acknowledgments

We wish to thank the anonymous reviewers for their helpful and instructive comments. This work has been supported by the Research Network MAVIR (S-0505/TIC-0267), the Regional Ministry of Education of the Community of Madrid, and by the Spanish Ministry of Science and Innovation project QEAVis-Catiex (TIN2007-67581-C02-01).

References

- B. E. Boser, I. Guyon and V. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the 5th Annual Workshop on computational Learning Theory*.
- C. Campbell. 2000. Algorithmic Approaches to Training Support Vector Machines: A Survey *Proceedings of ESANN'2000, European Symposium on Artificial Neural Networks*.
- O. Chapelle, M. Chi y A. Zien 2006. A Continuation Method for Semi-supervised SVMs. *Proceedings of ICML'06, the 23rd International Conference on Machine Learning*.
- O. Chapelle, V. Sindhwani, S. Keerthi 2008. Optimization Techniques for Semi-Supervised Support Vector Machines. *J. Mach. Learn. Res.*.
- C. Cortes and V. Vapnik. 1995. Support Vector Network. *Machine Learning*.
- C.-H. Hsu and C.-J. Lin. 2002. A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions on Neural Networks*.
- T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with many Relevant Features. *Proceedings of ECML98, 10th European Conference on Machine Learning*.
- T. Joachims. 1999. Transductive Inference for Text Classification Using Support Vector Machines. *Proceedings of ICML99, 16th International Conference on Machine Learning*.
- J. Kivinen and E.J. Smola and R.C. Williamson. 2002. Learning with Kernels.
- T. Mitchell. 1997. *Machine Learning*. McGraw Hill.
- H.-N. Qi, J.-G. Yang, Y.-W. Zhong y C. Deng 2004. Multi-class SVM Based Remote Sensing Image Classification and its Semi-supervised Improvement Scheme. *Proceedings of the 3rd ICMLC*.
- X. Qi and B.D. Davison. 2007. Web Page Classification: Features and Algorithms. *Technical Report LU-CSE-07-010*.
- B. Schölkopf and A. Smola. 1999. *Advances in Kernel Methods: Support Vector Learning*. MIT Press.
- F. Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, pp. 1-47.
- M.P. Sinka and D.W. Corne. 2002. A New Benchmark Dataset for Web Document Clustering. *Soft Computing Systems*.
- C.M. Tan, Y.F. Wang and C.D. Lee. 2002. The Use of Bigrams to Enhance Text Categorization. *Information Processing and Management*.
- J. Weston and C. Watkins. 1999. Multi-class Support Vector Machines. *Proceedings of ESAAN, the European Symposium on Artificial Neural Networks*.
- L. Xu y D. Schuurmans. 2005. Unsupervised and Semi-supervised Multiclass Support Vector Machines. *Proceedings of AAAI'05, the 20th National Conference on Artificial Intelligence*.
- Z. Xu, R. Jin, J. Zhu, I. King and M. R. Lyu. 2007. Efficient Convex Optimization for Transductive Support Vector Machine. *Advances in Neural Information Processing Systems*.
- Y. Yajima and T.-F. Kuo. 2006. Optimization Approaches for Semi-Supervised Multiclass Classification. *Proceedings of ICDM'06 Workshops, the 6th International Conference on Data Mining*.