International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 12, No. 1, pp. 40 – 47, 2015

A CROSS TRAINING CORRECTIVE APPROACH FOR WEB PAGE CLASSIFICATION

BELMOUHCINE Abdelbadie[1] And BENKHALIFA Mohammed[2]

Computer Science Laboratory (LRI), Computer science department,
Faculty of science, Mohammed V-Agdal University
Rabat, Morocco
[1]belmouhcine@gmail.com, [2]khalifa@fsr.ac.ma

Textual document classification is one challenging area of data mining. Web page classification is a type of textual document classification. However, the text contained in web pages is not homogenous since a web page can discuss related but different subjects. Thus, results obtained by a textual classifier on web pages are not as better as those obtained on textual documents. Therefore, we need to use a method to enhance results of those classifiers or more precisely a technique to correct their results. One category of techniques that address this problem is to use the test set hidden underlying information to correct results assigned by a textual classifier. In this paper, we propose a method that belongs to this category. Our method is a Cross Training based Corrective approach (CTC) for web page classification that learns information from the test set in order to fix classes initially assigned by a text classifier on that test set. This adjustment leads to a significant improvement on classification results. We tested our approach using three traditional classification algorithms: Support Vector Machine (SVM), Naïve Bayes (NB) and K Nearest Neighbors (KNN), on four subsets of the Open Directory Project (ODP). Results show that our collective and corrective approach, when applied after SVM, NB or KNN, enhances their classification results by up to 12.39%.

Keywords: corrective approach; web page classification; knn; svm; naïve bayes

1. Introduction

Document classification is the process of assigning classes to documents. The standard approach is to use a classifier such as Naïve Bayes, Support Vector Machine or K-Nearest Neighbors, to build a model based on manually labeled documents. Then, when test data are presented to the classifier, it uses that model to predict a category for each item in the test set without considering other items in it.

Many methods have used other items in test data when classifying a target item by exploiting some correlation between items. Three types of correlations have been used (Sen et al. 2008):

- The correlation between the class of item i, and features of i.
- The correlation between the class of item i, and features of items in the neighborhood of i.
- The correlation between the class of item i, and unobserved classes of items in the neighborhood of i.

In this paper, we present a corrective approach that splits the test data to n equal parts. Then, for each part, it rectifies test documents labels using a Cross Training scheme. For each part, it uses the n-1 remaining parts to train the classifier and applies it to fix categories assigned to items of the target part.

The originality of our proposed method resides in using the correlation between predicted labels and documents attributes to adjust already assigned categories.

We test our approach on four binary classification datasets extracted from the Open Directory Project (ODP) (« ODP - Open Directory Project » 2013).Our study shows that our approach ameliorates the classification's results thanks to the use of the underlying information present in test data and the correlation between attributes and predicting labels, to adjust classes assigned initially by the classifier.

In this paper, we propose a post classification corrective approach called Cross Training Correction (CTC). This approach is inspired from the k-fold cross validation (F. Mosteller et J. W. Tukey) technique and uses the hidden information present in the test set and the correlation between predicted labels and attributes, in order to make categories rectifications for classification's results improvement.

The rest of this paper is organized as follows. In section 2, we review recent work related to the subject. In section 3, we show our method in detail. In section 4, we present the experimental setting adopted. Then, we discuss obtained results. Finally, we conclude our work and cite some of our future perspectives.

2. Related Work

The use of ensemble of classifiers has been shown to give more accurate results than individual classification. In ensemble classification, many classifiers are built, and the final classification decision for each instance is made based on a form of voting. Many ensemble classifiers have been proposed in the literature, Bagging (Breiman 1996), Boosting (Freund et Schapire 1996) and Stacking (Wolpert 1992) are the best known techniques of ensemble classification. Liu et al. (Liu et al.) used stacking in the context of workforce classification. Stacking is a meta-learner based on cross validation used in order to obtain unbiased estimations on training data. First, they split the training set into J equal sized disjoint sets, and for each subset they train the classifier using other subsets and apply it to obtain predictions on the target subset. Then, given the predicted class for each example, they add it to the feature vector representing the example. Likewise, Kou and Cohen (Kou et Cohen 2007) introduced a collective classification approach based on stacking, called stacked graphical learning which is an approach that uses a base classifier to classify an instance using its features along with predictions of its related instances. Rather than using the same classifier for all inference iterations, they use a classifier per iteration. To avoid bias of using the same data for training and prediction, they use a cross validation-like approach to predict labels using the $(k-1)^{th}$ classifier in order to train the k^{th} classifier. They reported that their proposed learning method gave a good performance. In this paper, we propose a corrective approach based on cross validation

that does not use predicted labels to augment the feature vector, but instead uses them to train the classifier used for correction.

3. Proposed approach

In this paper, we propose a textual classification's corrective approach that can be applied in the context of web page classification. This method corrects results obtained by a text classifier using the underlying information hidden in the test set. It proceeds in n iterations, where n is simply the number of test set's splits adopted in our experiments. Let D= {(x, y)} be the dataset, where the pair (x, y) is an instance, x is the features vector of the instance and y={0,1} is the class of the instance. First, we divided our dataset to two distinct parts: a training set noted D_{Tr} and a test set noted D_T where $D_{TR} \cup D_T=D$ Then, we begin the bootstrapping step, in which a classifier is trained using instances belonging to D_{Tr} to predict labels of instances belonging to D_T. In the adjustment step, which contains n iterations, we split D_T to n equal parts in order to obtain n distinct sets D_i, i=1, 2..., n where $D_1 \cup D_2 \cup \ldots \cup D_n = D_T$. After splitting, our method uses a cross-validation like sliding window of size w=t/n (t=$|D_T|$). In each iteration k (k=1, 2..., n), the sliding window contains web pages of Dk. Then it uses $D_T \backslash D_k$ to train a classifier using predicted labels so that this latter can catch some useful underlying information from the test set that was not present in the training data. Then, it applies the classifier on the data in D_k (contained by the sliding window), to correct their classes. Finally, the window moves to the next k-subset of D_T and the process is reiterated until the window scans all data in D_T. The whole correction step of the method is repeated until convergence is obtained, or maximum number of iterations is reached. Figure 1 gives a summary of our corrective approach.

4. Experiment results

4.1. *Pre-processing*

We applied a number of pre-processing techniques to each web page in the dataset. The aim of those techniques is cleaning and normalizing the raw text contained in these web pages. In tokenization step we turn all terms to lower case, we removed some special characters, punctuation marks and numbers. Also, we removed all scripts, styles, mimes headings and HTML tags. For stemming process, we applied the well-known Porter method (Porter 1997). After the pre-processing stage, we build the dictionary that consists of words resulting from pre-processing. Thus, we consider web pages as bags of words. We represent our web pages using the conventional Vector Space Model (Salton et McGill 1986). We map each web page p onto its vector $v_p=(n_{1p},n_{2p},...,n_{mp})$; where n_{ip} denotes the weight of the i^{th} term in the web page p. We adopted the Term Frequency-Inverse Document Frequency (TF-IDF) (Salton et Buckley 1988; Jones 1972) based weighting model to obtain the weights.

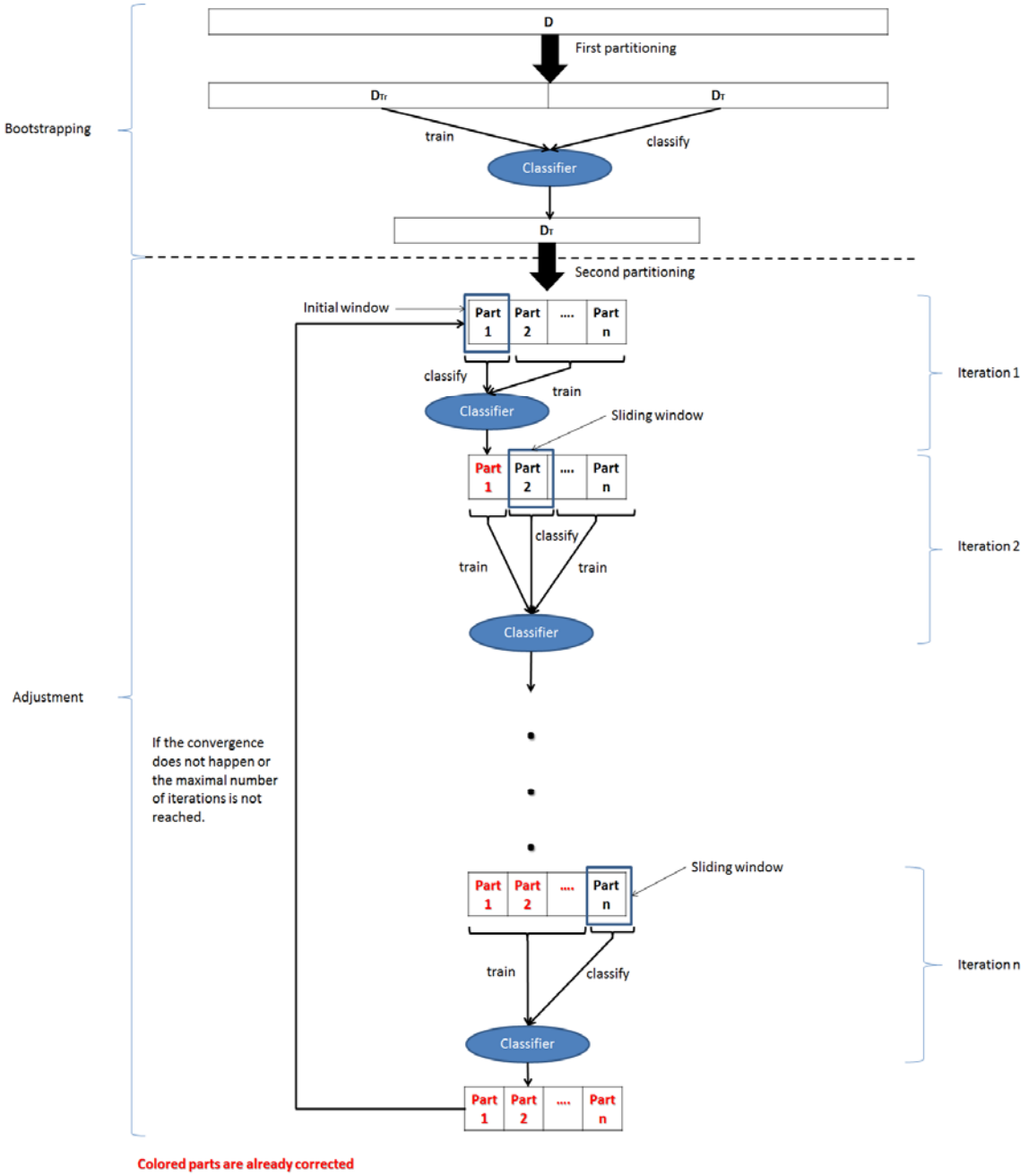

Figure 1 The corrective approach

4.2. *Classifiers used*

4.2.1. *Support Vector Machine*

We used our approach with the Support Vector Machines (SVM) (Cortes et Vapnik 1995) which is a powerful learning algorithm that works well in text classification (Joachims 1998). It is based on the Structured Risk Maximization theory and aims at minimizing the generalization error instead of the experimental error on training data alone. From multiple versions of SVM described in (Lin 2002), we used Sequential Minimal Optimization version which was developed in (Platt 1998; Keerthi et al. 2001). We used C=1 for the tolerance degree to errors. We also used a linear kernel, which proves to be efficient for text categorization, where we have high feature vector dimension (Joachims 1998).

4.2.2. *Naïve Bayes*

We also tested our approach with the Naïve Bayes (NB) which is a simple and very known classification algorithm (Mitchell 1997; McCallum et Nigam 1998). It uses the joint probabilities of attributes and classes to estimate the probabilities of categories given a document, and makes the assumption that features are conditionally independent of each other to make the computation of joint probabilities simple.

4.2.3. *K-Nearest Neighbors*

We used also our approach using the K Nearest Neighbors (KNN) which is the most simple classification algorithm (Aha et Kibler 1991). It is a type of lazy learners or instance based learners that predicts the category of an instance based on its K nearest training examples in the feature space based on an inter-instance similarity. This algorithm does not generate a model from training instances but rather stores all those training examples directly and uses them to determine the class of a new instance. To determine the suitable k parameter, we conduct a cross-validation on the training set.

4.3. *Datasets*

We test our approach using four binary problems, but our method can easily be extended to multi-label classification problems by applying “one against others” classification.
Datasets used in this paper are taken from the Open Directory Project (ODP) (« ODP - Open Directory Project » 2013) which is a tremendous repository containing around 4.6 million web pages and is organized into 765,282 categories and subcategories (Henderson 2009). We constructed four binary classification tasks: “Adult” vs. “Other” (1606 web pages), “KidsAndTeens” vs. “Other” (1591 web pages), “Health” vs. “Other” (1749 web pages), “Games” vs. “Other” (2012 web pages).

Our approach needs some web pages to train the content based classifier and some web pages to test the method. We conduct all our experiments using 10 fold cross validation (F. Mosteller et J. W. Tukey) in order to reduce the uncertainty of data split between

training and test data. We used one fold for training and the nine others for testing so that the number of unlabeled web pages be widely greater than labeled ones.

4.4. *Evaluation measures*

To evaluate results of our approach, we used the standard metrics: recall, precision and F1, which are commonly used to evaluate the classification task. Recall is defined to be the proportion of correct assignments by the system within the total number of correct assignments. Precision is the proportion of correct assignments by the system within the total number of the system's assignments. F1, introduced by Van Rijsbergen (Rijsbergen 1979) is the equally weighted average of recall and precision.

5. Results and discussion

From Table 1, which contains results obtained with a number of splits equal to 10, we can observe that our approach ameliorates results of the three base classifiers KNN, SVM and Naïve Bayes for almost all datasets used. This proves that the correlation between predicted labels and web pages attributes helps ameliorate classification results. Exceptionally, our proposed approach does not increase performances when using KNN on Health dataset. This due to the recall of KNN on Health dataset is very low (0.365). This means that lot of instances belonging to Health category, are classified as Other by KNN. Thus, classifier used for correction suffers from many noises created by wrong dependencies between labels and data.

Table 1. Performance of our approach contrasted to standard approaches

	Adults			**KidsAndTeens**			**Health**			**Games**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
KNN	0.623	0.682	0.651	0.546	0.712	0.618	0.718	0.365	0.484	0.466	0.885	0.611
KNN+CTC	0.642	0.8	0.712	0.571	0.795	0.665	0.703	0.323	0.443	0.53	0.962	0.683
NB	0.892	0.803	0.845	0.67	0.739	0.703	0.901	0.89	0.895	0.818	0.849	0.833
NB+CTC	0.986	0.806	0.887	0.684	0.741	0.711	0.916	0.873	0.894	0.821	0.851	0.836
SVM	0.833	0.805	0.819	0.67	0.703	0.686	0.93	0.937	0.933	0.856	0.92	0.887
SVM+CTC	0.871	0.868	0.869	0.76	0.783	0.771	0.957	0.95	0.953	0.906	0.921	0.913

6. Conclusion

In this paper, we have proposed a new approach that helps improve results given by a text classifier. This method takes benefit from underlying hidden information present in the test set to collectively adjust categories of web pages. Our experiments show that within an appropriate empirical setting, our approach improves performance of three traditional classifiers: SVM, Naïve Bayes, and KNN. Our main findings include:

(1) The use of the correlation between predicted labels and web pages helps adjusting the classes assigned initially by the textual classifier.
(2) Results obtained after the initial classification have an influence on the performance of our corrective approach.

We hope that our corrective approach will help future researchers to conduct further studies on classification's results correction.

In the future, we will augment our corrective method by the use of contextual information contained in web pages to get more performances after the correction.

References

Aha, D., et D. Kibler. 1991. « Instance-based learning algorithms ». *Machine Learning* 6: 37-66.

Breiman, Leo. 1996. « Bagging Predictors ». *Machine Learning* 24 (2): 123-40. doi:10.1023/A:1018054314350.

Cortes, Corinna, et Vladimir Vapnik. 1995. « Support-Vector Networks ». *Mach. Learn.* 20 (3): 273-97. doi:10.1023/A:1022627411411.

F. Mosteller, et J. W. Tukey. « Data Analysis, Including Statistics ». In *Handbook of Social Psychology (G. Lindzey and E. Aronson, eds.)*, 2e éd., 2:80-203. Addison-Wesley, Reading, MA.

Freund, Yoav, et Robert E. Schapire. 1996. *Experiments with a New Boosting Algorithm*.

Henderson, Lachlan. 2009. « Automated Text Classification in the DMOZ Hierarchy ».

Joachims, Thorsten. 1998. « Text categorization with Support Vector Machines: Learning with many relevant features ». In *Machine Learning: ECML-98*, édité par Claire Nédellec et Céline Rouveirol, 137-42. Lecture Notes in Computer Science 1398. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/BFb0026683.

Jones, Karen Spärck. 1972. « A statistical interpretation of term specificity and its application in retrieval ». *Journal of Documentation* 28: 11-21.

Keerthi, S. S., S. K. Shevade, C. Bhattacharyya, et K. R. K. Murthy. 2001. « Improvements to Platt's SMO Algorithm for SVM Classifier Design ». *Neural Computation* 13 (3): 637-49. doi:10.1162/089976601300014493.

Kou, Zhenzhen, et William W. Cohen. 2007. « Stacked graphical models for efficient inference in markov random fields ». In *In Proceedings of the 2007 SIAM International Conference on Data Mining*.

Lin, Chih-Jen. 2002. « Asymptotic convergence of an SMO algorithm without any assumptions ». *IEEE Transactions on Neural Networks* 13 (1): 248-50. doi:10.1109/72.977319.

Liu, Yan, Zhenzhen Kou, Claudia Perlich, et Richard Lawrence. *Intelligent System for Workforce Classification*.

McCallum, Andrew, et Kamal Nigam. 1998. *A comparison of event models for Naive Bayes text classification*.

Mitchell, Tom M. 1997. *Machine Learning*. 1re éd. McGraw-Hill Science/Engineering/Math.

« ODP - Open Directory Project ». 2013. Consulté le février 24. http://www.dmoz.org/.

Platt, John C. 1998. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING.

Porter, M. F. 1997. « Readings in information retrieval ». In , édité par Karen Sparck Jones et Peter Willett, 313-16. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=275537.275705.

Rijsbergen, C. J. Van. 1979. *Information Retrieval*. 2nd éd. Butterworth-Heinemann.

Salton, Gerard, et Christopher Buckley. 1988. « Term-weighting approaches in automatic text retrieval ». *Inf. Process. Manage.* 24 (5): 513-23. doi:10.1016/0306-4573(88)90021-0.

Salton, Gerard, et Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.

Sen, Prithviraj, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, et Tina Eliassi-rad. 2008. *Collective classification in network data*.

Wolpert, David H. 1992. « Stacked Generalization ». *Neural Networks* 5: 241-59.