On the Class Distribution Labelling Step Sensitivity of CO-TRAINING

Edson T. Matsubara, Maria C. Monard, and Ronaldo C. Prati

Department of Computer Science
ICMC/USP - São Carlos
Laboratory of Computational Intelligence - LABIC
P.O. Box 668
13560-970 São Carlos,SP, Brazil.
{edsontm,mcmonard,prati}@icmc.usp.br

Abstract. CO-TRAINING can learn from datasets having a small number of labelled examples and a large number of unlabelled ones. It is an iterative algorithm where examples labelled in previous iterations are used to improve the classification of examples from the unlabelled set. However, as the number of initial labelled examples is often small we do not have reliable estimates regarding the underlying population which generated the data. In this work we make the claim that the proportion in which examples are labelled is a key parameter to CO-TRAINING. Furthermore, we have done a series of experiments to investigate how the proportion in which we label examples in each step influences CO-TRAINING performance. Results show that CO-TRAINING should be used with care in challenging domains.

1 Introduction

Semi-supervised learning uses a set of examples where only a few examples are labelled, and the goal is to predict the labels of the remaining unlabelled examples. The main idea of semi-supervised learning is to investigate ways whereby using the unlabelled data it is possible to effectively improve classification performance, compared with a classifier build only using the labelled data, *i.e.* without considering the unlabelled data. For these reasons, semi-supervised learning is considered as the middle road between supervised and unsupervised learning.

Methods that have been proposed under this paradigm include the multi-view semi-supervised CO-TRAINING method (1), dealt with in this work. CO-TRAINING applies to datasets that have a natural separation of their attributes into at least two disjoint sets, so that there is a partitioned description of each example into each distinct view. For each view, the set of few labelled examples is given to learning algorithms to induce independent classifiers. Each classifier is used to classify the unlabelled data in its respective view. Afterwards, examples which have been classified with a higher degree of confidence for all views are included in the set of labelled examples and the process is repeated using the augmented labelled set until a stop criterion is met. However, due to the limited number of initial training examples available in semi-supervised learning, it is not possible to estimate the class distribution of the dataset in advance. Furthermore, when examples are labelled, as there is no information concerning class distribution, we do not know in which class proportion the higher confidence labelled examples should be included in the set of labelled examples in each iteration. This is a question of practical importance, and in this work we analyse the effect of class distribution in CO-TRAINING. Experimental results of CO-TRAINING performance with respect to accuracy, number of incorrectly labelled examples and AUC show that, although the best results are obtained if the true class distribution of the examples is known, for some domains where there is a great separability among classes the performance of CO-TRAINING can also be competitive when this information is not available. However, CO-TRAINING should be used with caution in challenging domains.

The rest of this work is organised as follows: Section 2 presents related work on semi-supervised learning. Section 3 describes CO-TRAINING. Section 4 discusses the class distribution sensitivity problem. Section 5 reports the experimental results, and Section 6 concludes the work.

2 Related Work

Semi-supervised learning algorithms can be divided into single-view and multi-view (2; 3). In a single-view scenario the algorithms have access to the entire set of domain attributes. Single-view algorithms can be split up into transductives (4), Expectation Maximization (EM) variations (5), background knowledge based algorithms (6) and graph-based methods (3). In a multi-view setting, the attributes are presented in subsets (views) which are sufficient to learn the target concept. Multi-view algorithms are based on the assumption that the views are both *compatible* (all examples are labelled identically by the target concepts in each view), and *uncorrelated* (given the label of any example, its descriptions in each view are independent)

The CO-TRAINING algorithm provides the basis for multi-view learning. Following CO-TRAINING some multi-view learning algorithms have been proposed, such as: CO-EM (7) which combines EM and CO-TRAINING; CO-TESTING (2) which combines active and semi-supervised learning, and CO-EMT (2) an extension of CO-TESTING with CO-EM. The use of Support Vector Machines (SVM) instead of *Naive Bayes* (NB) as the base-learning learner is proposed in (8). An improved version of CO-EM using SVM is proposed in (9) showing experimental results that outperform other algorithms. CO-TRAINING requires the instance space to be described with sufficient and redundant views. On the other hand, the TRI-TRAINING algorithm (10) neither requires this nor imposes any constraints on the supervised learning algorithm; its applicability is broader than previous CO-TRAINING style algorithms. The majority of these applications and related work barely consider the class distribution.

3 The CO-TRAINING Algorithm

Given a set of N examples $E = \{E_1, ..., E_N\}$ defined by a set of M attributes $\mathbf{X} = \{X_1, X_2, ..., X_M\}$ and the class attribute Y, where we only know the class attribute for a few examples, CO-TRAINING needs at least two disjoint and compatible views D_1 and D_2 of the set of examples E to work with. In other words, for each example $j = 1, 2...N$ in D_1 we should have its j-th counterpart (compatible example) in D_2. We shall refer to these two views as $\mathbf{X}_{D_1}$ and $\mathbf{X}_{D_2}$ such that $\mathbf{X} = \mathbf{X}_{D_1} \cup \mathbf{X}_{D_2}$ and $\mathbf{X}_{D_1} \cap \mathbf{X}_{D_2} = \emptyset$. Furthermore, the set of labelled examples in each view should be adequate for learning.

Set E can be divided into two disjoint subsets L (Labeled) and U (Unlabelled) of examples. Both subsets L and U are further divided into two disjoint views respectively called, L_{D_1}, L_{D_2} and U_{D_1}, U_{D_2}. These four subsets L_{D_1}, L_{D_2}, U_{D_1} and U_{D_2}, illustrated in Figure 1, as well as the maximum number of iterations k, constitute the input of CO-TRAINING described by Algorithm 1.

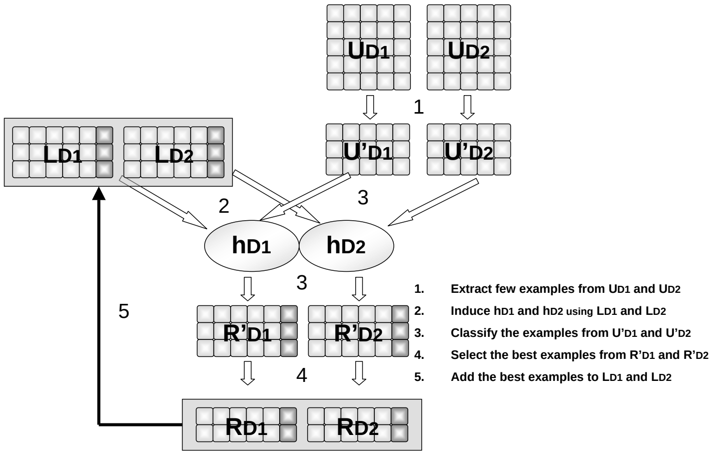
UD1
UD2
1
U'D1
U'D2
LD1
LD2
2
3
hD1
hD2
3
5
R'D1
R'D2
4
RD1
RD2
1. Extract few examples from UD1 and UD2
2. Induce hD1 and hD2 using LD1 and LD2
3. Classify the examples from U'D1 and U'D2
4. Select the best examples from R'D1 and R'D2
5. Add the best examples to LD1 and LD2

Fig. 1. CO-TRAINING

Initially, two small pools U'_{D_1} and U'_{D_2} of compatible unlabelled examples, withdrawn from U_{D_1} and U_{D_2} respectively, are created, and the main loop of Algorithm 1 starts. First, the sets of training examples L_{D_1} and L_{D_2} are used to induce two classifiers h_{D_1} and h_{D_2}, respectively. Next, the set of examples U'_{D_1} is labelled using h_{D_1} and inserted in R'_{D_1}, and the set of examples from U'_{D_2} is labelled using h_{D_2} and inserted in R'_{D_2}. Both sets of labelled examples are given to the function *bestExamples* which is responsible for ranking compatible examples from R'_{D_1} and R'_{D_2} that have the same class label prediction, and

Algorithm 1: CO-TRAINING

```
Input: $L_{D_1}, L_{D_2}, U_{D_2}, k$
Output: $L_{D_1}, L_{D_2}$
Build $U'_{D_1}$ and $U'_{D_2}$ as described;
$U_{D_1} = U_{D_1} - U'_{D_1}$;
$U_{D_2} = U_{D_2} - U'_{D_2}$;
for $i = 0$ to $k$ do
    Induce $h_{D_1}$ from $L_{D_1}$;
    Induce $h_{D_2}$ from $L_{D_2}$;
    $R'_{D_1} = h_{D_1}(U'_{D_1})$ set of classified examples from $U'_{D_1}$;
    $R'_{D_2} = h_{D_2}(U'_{D_2})$ set of classified examples from $U'_{D_2}$;
    $(R_{D_1}, R_{D_2}) = bestExamples(R'_{D_1}, R'_{D_2})$;
    $L_{D_1} = L_{D_1} \cup R_{D_1}$;
    $L_{D_2} = L_{D_2} \cup R_{D_2}$;
    if $U_{D_1} = \emptyset$ then return$(L_{D_1}, L_{D_2})$ else
        Randomly select compatible examples from $U_{D_1}$ and $U_{D_2}$ to replenish
        $U'_{D_1}$ and $U'_{D_2}$ respectively;
    end
end
return$(L_{D_1}, L_{D_2})$;
```

selecting from them the "best" pairs of compatible examples to be inserted in L_{D_1} and L_{D_2} respectively. After that the process is repeated until a stop criterion is met — either the maximum number of iterations defined by the user or the set U_{D_1} (or its counterpart U_{D_2}) is empty.

Algorithm 1 describes the general idea of CO-TRAINING using the same base-learning learning algorithm (*Naive Bayes* in the original proposal) which makes it possible to construct a third classifier from h_{D_1} and h_{D_2} called combined classifier (1). Furthermore, Algorithm 1 only uses two visions and binary class datasets. However, as suggested by its authors, there are several features that can be included in the original version. Our implementation of CO-TRAINING includes several such features which enable us to test its behavior under different situations. These features include: more than two visions; more than two classes; variable number of examples and proportion of examples by class in the initial labelled sets L_{D_i} as well as sets U'_{D_i}; different base-learning algorithms; maximum number of "best" classified examples in each class that can be inserted in L_{D_i} during each iteration, and others.

4 Class proportion labelling sensitivity of CO-TRAINING

A common assumption in the design of standard learning algorithms is that training examples are drawn from the same underlying distributions the model is expected to make predictions. In CO-TRAINING, though, this assumption does not hold because the training set of examples is growth while the algorithm is running, and the amount of labelled examples, as well as the proportion in which examples are labelled, is generally a parameter of the algorithm set by the user.

For example, suppose we are using CO-TRAINING to label data for web page classification. In a typical application, we construct a robot crawler that visits some web sites and downloads all pages of interest. We then ask a human expert to hand label some web pages with the classes we are interested in. As we generally do not know how many examples should be labelled for each class, a fair option is to ask the expert to label an even number of examples for each class. Another option is to draw a small sample of examples and ask the expert to label this sample. Although one may argue that the latter option would produce a more reliable estimate of the class distribution than the former, this is not necessarily true as the crawler might have some bias when retrieving web pages. Thus, in both cases we do not have a good estimate of which proportion we should label examples in each CO-TRAINING iteration.

As CO-TRAINING is an iterative process, where examples labelled in previous iterations are used to build models to label new data, in this work we argue that the proportion in which examples are labelled is a key parameter of the CO-TRAINING algorithm. The main point is that we may not know beforehand the true underlying distribution we should use as a parameter for CO-TRAINING beforehand. As the base-classifier might be sensitive to class skews, feeding the algorithm with a class distribution different from the true one would bias the base-classifier used by CO-TRAINING towards an inaccurate classifier. As a consequence, the number of examples incorrectly labelled would increase, degrading the performance of CO-TRAINING.

Although it is very difficult to characterize the effect that changing class distribution would have in learning algorithms, several studies evaluate its behaviour for a number of well-known algorithms. (11) conducts an extensive experimentation using the decision tree algorithm C4.5 with datasets sampled under several different class distributions. The authors conclude that, on average, the natural class distribution produces the most accurate classifiers. (12) claims that when the independence assumption of attributes is violated, the Naive Bayes algorithm is affected by changing class distributions. The author shows that this sensitivity also holds for other algorithms, such as logistic regression and hard margin SVMs. (13) further extends these results claiming that the sensitivity could not only be attributed to the learning system but also to the dataset at hand. As CO-TRAINING uses learning algorithms as base-classifiers, this sensitivity is automatically inherited from the learning system. The next section shows how this sensitivity affects the results for the datasets used in our experiments.

5 Experimental Evaluation

We carried out an experimental evaluation using three different text datasets: a subset of the UseNet news articles (20-NewsGroups) (14); abstracts of academic papers, titles and references collected from *Lecture Notes in Artificial Intelligence* (LNAI) (15) and links and web pages from the COURSE dataset (1).

For the first dataset we created a subset of the 20-newsgroups selecting 100 texts from `sci.crypt`, `sci.electronics`, `sci.med`, `sci.space`, `talk.politics.guns`, `talk.politics.mideast`, `talk.politics.misc` and `talk.religion.misc`. All texts from the first 4 newsgroups were labelled as `sci` (400 - %50) and texts from the remaining newsgroups were labelled as `talk` (400 - %50). The LNAI dataset contains 396 papers from *Case Based Reason* (277 - 70%) and *Inductive Logic Programming* (119 - 30%). The COURSE dataset[1] consists of 1051 web pages collected from various Computer Science department web sites, and divided into several categories. This dataset already provides the two views for each web page example. One view consists of words appearing on the page, and the other view consists of the underlined words from other pages which point to the web page. However, analysing the examples in the original dataset, we found 13 examples which are either empty (no text) or its compatible example in the counterpart view is missing. Thus, the original dataset was reduced to 1038 examples. Similar to (1), web pages were labelled as course (221 - 20%), and the remaining categories as non-course (817 - 80%).

Using PRETEXT [2], a text pre-processing tool we have implemented (16), all text datasets were decomposed into the attribute value representation using the bag-of-words approach. Stemming and Luhn cut-offs were also carried out. For datasets NEWS and LNAI the two views were constructed following the approach we proposed in (17), using *1-gram* representation as one view and *2-gram* as the second view of the datasets. For the *2-gram* view in the NEWS dataset, the minimum Luhn cut-off was set to 3. For the remaining views, the minimum Luhn cut-off was set to 2. The maximum Luhn cut-offs were left unbounded. For dataset COURSE *1-gram* was used in both views, named TEXT and LINKS. Table 1 summarises the datasets used in this work. It shows the dataset name (Dataset); number of documents in the dataset (#Doc); number of generated stems (#Stem); number of stems left after performing Luhn cut-offs in each view (#Attributes), and class distribution (%Class).

As all datasets are completely labelled, we can compare the labels assigned by CO-TRAINING in each iteration with the true labels of the datasets. In other words, we use CO-TRAINING in a simulated mode, in which the true labels are hidden from the algorithm and are only used to measure the number of examples wrongly labelled by CO-TRAINING. In our experiments we used *Naive Bayes* (NB) as a CO-TRAINING base-classifier. In order to obtain a lower bound of the error that CO-TRAINING can reach on these datasets, we measured the error

[1] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/`

[2] `http://www.icmc.usp.br/~edsontm/pretext/pretext.html`

rate of NB using all labelled examples using 10-fold cross-validation. Results (mean error and respective standard deviation) are shown in the last column (NB Error) of Table 1.

Dataset	#Doc	View	#Stem	#Attr.	Class	%Class	NB Error	Overall Error
news	800	1-gram	15711	8668	`sci`	50%	2.5 (1.7)	
					`talk`	50%	0.8 (1.2)	1.6 (1.0)
		2-gram	71039	4521	`sci`	50%	2.0 (2.0)	
					`talk`	50%	0.5 (1.1)	1.3 (1.2)
lnai	396	1-gram	5627	2914	ilp	30%	1.7 (3.7)	
					cbr	70%	1.4 (1.9)	1.5 (1.8)
		2-gram	21969	3245	ilp	30%	1.8 (1.7)	
					cbr	70%	1.5 (1.9)	1.8 (1.7)
course	1038	text	13198	6870	`course`	20%	16.3 (5.4)	
					`non-course`	80%	3.8 (2.0)	6.5 (2.3)
		links	1604	1067	`course`	20%	9.6 (7.6)	
					`non-course`	80%	16.0 (4.7)	14.6 (3.5)

Table 1. Datasets description and *Naive Bayes* error

To assess the behaviour of co-training using cross-validation, we adapted the sampling method as follows: first, the examples in both views are paired and marked with an ID. Then, we sample the folds so that both training and test samples are compatible, *i.e.*, an example marked with a given ID appears only in the training or test sample in both views.

All experiments were carried out using the same number of initial labelled examples (30 examples) evenly distributed by class (50% - 50%). In each iteration, up to 10 “best” examples were allowed to be labelled. Furthermore, to analyse the impact of the class distribution we varied the number of examples in each class. We used 0.6 as a threshold to select the best examples, *i.e.* compatible candidates must have been labelled by NB with a probability greater than 0.6.

Table 2 shows the mean value and standard deviation of results obtained using 10-fold cross validation. The first line indicates the maximum number of examples by class that can be labelled in each iteration: `sci/talk` for news, ilp/cbr for lnai and `course/non-course` for course dataset. For each dataset the first four lines show the number of examples in each class that have been wrongly (W) or rightly (R) labelled; LSize is the number of examples labelled by co-training, including the 30 initial examples; USize is the number of unlabelled examples left; Error and AUC are respectively the error rate and the area under the ROC curve of the combined classifier, and Wrong is the total number of examples wrongly labelled. The best mean results for these last three measures are in bold.

For all datasets co-training ended due to reaching the condition of an empty set of unlabelled examples in iterations 64, 28 and 86 for datasets news, lnai and course respectively. As can be observed, best results for news and

COURSE datasets are obtained whenever examples are labelled considering the dataset distribution (5/5 for NEWS and 2/8 for COURSE). For LNAI dataset, although the best result is not obtained for its exact proportion 3/7, it is obtained by its similar proportion 2/8. For this dataset, labelling examples using a slight biased proportion towards the minority and most error-prone class (see Table 1) seems to improve classification. In both cases the total number of labelled examples is the same (LSize $\simeq$ 300). The main difference is in the error of each class: while 3/7 proportion labels all CBR examples correctly, 2/8 proportion labels all ILP examples correctly.

Moreover, for the best results the mean error rate of the combined classifiers are compatible with the once obtained using the labelled examples (Table 1), although the COURSE dataset presents a far greater variance.

	2/8	3/7	5/5	7/3	8/2
NEWS dataset					
sci(W)	18.00 (26.45)	10.60 (15.47)	1.10 (1.85)	0.40 (0.52)	0.80 (0.42)
sci(R)	344.50 (2.72)	339.40 (2.50)	325.70 (11.51)	203.60 (0.52)	139.50 (1.51)
talk(W)	1.60 (1.17)	2.20 (0.63)	5.70 (10.03)	42.50 (30.34)	131.00 (18.89)
talk(R)	139.40 (1.17)	201.80 (0.63)	324.30 (10.03)	345.70 (1.89)	347.80 (3.08)
LSize	503.50 (26.53)	554.00 (15.30)	656.80 (9.77)	592.20 (30.07)	619.10 (17.00)
U'Size	206.50 (26.53)	156.00 (15.30)	53.20 (9.77)	117.80 (30.07)	90.90 (17.00)
Error	3.00 (3.24)	2.38 (3.70)	**1.88 (2.14)**	6.25 (5.14)	19.00 (3.53)
AUC	0.98 (0.02)	0.98 (0.03)	**0.99 (0.02)**	0.97 (0.04)	0.92 (0.05)
Wrong	19.80 (26.96)	12.80 (15.80)	**6.80 (11.77)**	43.70 (30.29)	133.50 (19.31)
LNAI dataset					
ilp(W)	0.00 (0.00)	1.30 (1.25)	5.40 (1.71)	9.30 (3.23)	12.30 (5.10)
ilp(R)	69.00 (0.00)	94.20 (2.20)	101.00 (1.49)	100.80 (1.14)	101.70 (1.57)
cbr(W)	0.70 (0.95)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
cbr(R)	230.30 (0.95)	204.00 (0.00)	150.00 (0.00)	96.00 (0.00)	69.00 (0.00)
LSize	300.00 (0.00)	299.50 (1.08)	256.40 (2.41)	206.10 (3.54)	183.00 (5.10)
U'Size	50.00 (0.00)	50.50 (1.08)	93.60 (2.41)	143.90 (3.54)	167.00 (5.10)
Error	**1.26 (1.33)**	2.02 (2.00)	2.03 (1.07)	3.28 (1.69)	4.80 (3.03)
AUC	**1.00 (0.00)**	1.00 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Wrong	**0.70 (0.95)**	1.30 (1.25)	5.60 (1.90)	9.30 (3.23)	12.50 (5.04)
COURSE dataset					
course(W)	34.40 (29.73)	103.90 (66.05)	252.30 (72.89)	423.40 (27.35)	434.80 (112.58)
course(R)	146.00 (26.82)	132.80 (27.26)	155.50 (13.34)	175.40 (6.00)	179.30 (10.89)
ncourse(W)	5.30 (3.13)	7.20 (8.00)	4.20 (4.59)	1.50 (2.92)	2.40 (3.34)
ncourse(R)	505.20 (154.07)	307.10 (227.37)	146.80 (110.20)	81.60 (31.65)	81.30 (56.98)
LSize	690.90 (150.92)	551.00 (186.16)	558.80 (49.82)	681.90 (23.39)	697.80 (66.62)
U'Size	239.10 (150.92)	379.00 (186.16)	371.20 (49.82)	248.10 (23.39)	232.20 (66.62)
Error	**14.11 (13.26)**	32.65 (20.15)	49.43 (15.95)	61.91 (8.07)	60.29 (17.28)
AUC	**0.92 (0.08)**	0.82 (0.11)	0.71 (0.09)	0.68 (0.07)	0.67 (0.07)
Wrong	**40.20 (31.71)**	112.80 (67.28)	258.70 (72.08)	429.80 (25.59)	442.60 (111.98)

Table 2. CO-TRAINING results for NEWS, LNAI and COURSE datasets

Analysing the behaviour of CO-TRAINING when changing the class distribution of labelled examples shows an interesting pattern. For the balanced dataset NEWS, skewing the proportion of labelled examples towards the `talk` class (*i.e*, labelling more examples from the `talk` class: 7/2 and 8/2) does not diminish the performance significantly. The other way dramatically increases the error rate (from 1.88 in 5/5 labelling to 19.00 in 8/2 labelling) as well as in the number of examples incorrectly labelled (6.8% to 133.50%). For the imbalanced datasets the picture is clearer. Both the error rate and the number of incorrectly labelled examples increase as we go towards the opposite direction in terms of proportion of labelled examples.

Another interesting result is related to the AUC. For the datasets with high AUC values — NEWS and LNAI —(near 1), the degradation in performance is weaker than for the COURSE dataset. This is because AUC values near 1 are a strong indication of a domain with a great separability, *i.e.*, domains in which the classes could be more easily separated from the others, and it is easy for the algorithm to construct accurate classifiers even if the proportion of examples in the training set is different from the natural one.

6 Conclusions and Future Work

In this work we analyse, for a fixed set of few labelled examples, the relationship between the unknown class distribution of domains and CO-TRAINING performance with respect to which proportion we should label examples in each iteration. Experimental results evaluated using the labelling accuracy, combined classifier error rate and AUC show that the best performance is achieved whenever we label examples in a proportion equal or close to the natural class distribution present in the datasets. Furthermore, labelling examples in proportions very different from the natural class distribution seems to decrease CO-TRAINING performance, especially in challenging domains. These results should be interpreted as a warning to anyone who is using CO-TRAINING for data labelling.

As future work, we are investigating ways to neutralise or overcome the class proportion labelling dependency of CO-TRAINING. (12) presents some methods aimed at correcting the class proportion when this proportion is not known in a classification context. It would be interesting to adapt this method to CO-TRAINING learning. A possible adaptation would be to label examples in the same proportion as the best examples appear in the L' set. This approach leads to labelling a flexible proportion of examples in each iteration and could bias the class distribution in the L set towards the natural one. However, experimental research should be carried out to analyse the feasibility of this approach.

Acknowledgements: The authors would like to thank FAPESP (Process 2005/03792-9) and CAPES, Brazil, for financial support.

References

[1] Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. 11th Annu. Conf. on Comput. Learning Theory, ACM Press, New York, NY (1998) 92–100

[2] Muslea, I.: Active Learning with Multiple Views (2002) PhD Thesis, University Southern California.

[3] Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005) http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

[4] Vapnik, V.: Statistical learning theory. John Wiley & Sons (1998)

[5] Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Conference on Information and Knowledge Management. (2000) 86–93

[6] Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proc. of the 18th Int. Conf. on Machine Learning. (2001) 577–584

[7] Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39** (2000) 103–134

[8] Kiritchenko, S., Matwin, S.: Email classification with co-training. Technical report, University of Otawa (2002)

[9] Brefeld, U., Scheffer, T.: Co-EM Support Vector Learning. In: Proc. of the Int. Conf. on Machine Learning, Morgan Kaufmann (2004) 16

[10] Zhou, Z.H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. In: IEEE Transactions on Knowledge and Data Engineering. Volume 17. (2005) 1529–1541

[11] Weiss, G.M., Provost, F.J.: Learning when training data are costly: The effect of class distribution on tree induction. J. Artif. Intell. Res. (JAIR) **19** (2003) 315–354

[12] Zadrozny, B.: Learning and evaluating classifiers under sample selection bias. In Brodley, C.E., ed.: Proc of the 21st Int. Conf. on Machine Learning (ICML 2004), ACM (2004) 114–121

[13] Fan, W., Davidson, I., Zadrozny, B., Yu, P.S.: An improved categorization of classifier's sensitivity on sample selection bias. In: Proc of the 5th IEEE Int. Conf. on Data Mining (ICDM 2005), IEEE Computer Society (2005) 605–608

[14] Blake, C., Merz, C.: UCI Repository of Machine Learning Databases (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html.

[15] Melo, V., Secato, M., Lopes, A.A.: Automatic extraction and identification of bibliographical information from scientific articles (in Portuguese). In: IV Workshop on Advances and Trend in AI, Chile (2003) 1–10

[16] Matsubara, E.T., Martins, C.A., Monard, M.C.: Pretext: A preprocessing text tool using the bag-of-words approach. Technical Report 209, ICMC-USP (2003) (in portuguese) ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/RT_209.zip.

[17] Matsubara, E.T., Monard, M.C., Batista, G.E.A.P.A.: Multi-view semi-supervised learning: An approach to obtain different views from text datasets. In: Advances in Logic Based Intelligent Systems. Volume 132., IOS Press (2005) 97–104