14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

IMAGE CLASSIFICATION USING LABELLED AND UNLABELLED DATA

Irena Koprinska[1], Da Deng[2] and Felix Feger[3]

[1]School of Information Technologies, University of Sydney, NSW 2006, Sydney Australia, email: irena@it.usyd.edu.au
[2]Department of Information Science, University of Otago, Dunedin, New Zealand, email: ddeng@infoscience.otago.ac.nz
[3]Otto Friedrich Universität Bamberg, Kapuzinerstraße 16, 96045 Bamberg, e-mail: Felix.Feger@stud.uni-bamberg.de

ABSTRACT

In this paper we present a case study of co-training to image classification. We consider two scene classification tasks: indoors vs. outdoors and animals vs. sports. The results show that co-training with Naïve Bayes using 8-10 labelled examples obtained only 1.2-1.5% lower classification accuracy than Naïve Bayes trained on the full labelled version of the training set (138 examples in task 1 and 827 examples in task 2). Co-training was found to be sensitive to the choice of base classifier, with Naïve Bayes outperforming Random Forest. We also propose a simple co-training modification based on the different inductive basis of classification algorithms and show that it is a promising approach.

1. INTRODUCTION

Image classification is an important problem in the area of content-based image and video retrieval. It involves mapping low level features (e.g. colour, edge, texture) to semantic categories (e.g. indoors/outdoors, anchorperson/no anchorperson, etc). To build an accurate classifier, a large number of examples labelled with their correct category are needed. Acquiring labelled examples is costly and time consuming as it requires human effort.

The co-training paradigm [1] tries to overcome this problem by taking advantage of the more abundant and easily available unlabeled data. It learns from a small set of labelled and a large set of unlabelled examples. The standard co-training requires that the data is naturally described by two disjoint feature sets, called views. There are two requirements for the views; they should be: 1) sufficiently strong, i.e. good accuracy can be achieved using each of them individually, and 2) conditionally independent given the class. Under these assumptions it was proven that a task that is learnable with random noise is learnable with co-training [1].

Co-training has been successfully applied in the area of text categorization. Blum and Mitchell [1] classified web pages as course and non-course home pages using the following two views: the words in the web page and the words in the hyperlinks pointing to the web page. Nigam and Ghani [2] investigated the effect of the dependence between the views. As most datasets do not come with natural split of features, which limits the applicability of co-training, a random split was tested. Co-training was found to work better on truly independent views than on random views. It was shown that if there is sufficient redundancy in data, the performance of co-training with random split is comparable to natural split. The experimental domain was classification of postings into newsgroups. Co-training was also applied for filling e-mails into folders [3] and noun phrase identification [4].

In this paper we present a case study of co-training to two image classification tasks. Previous research mainly concentrated on applying co-training for classification of text documents. Text categorization tasks are in fact less suitable for co-training because of the following reasons. First, most often they do not come with a natural split of the words into two sets which motivated the study of random split [2]. Second, even if a natural split exists (e.g. subject and body in e-mail classification), the words in the two sets are hardly independent, and most often one of the feature sets is not strong enough (e.g. the subject view due to its smaller vocabulary). Image classification, on the other hand, comes with natural split of features; for example, the two feature sets can be colour and edge, or local and global features. These feature sets are reasonably independent, and most often strong enough. This makes co-training particularly suitable for image classification.

We also propose a modification of the standard co-training algorithm motivated by the fact that different algorithms have different inductive bias. Finally, we study the learning behaviour of a new and powerful classifier: Random Forest. Previous research on co-training has mainly concentrated on using Naïve Bayes as base classifier.

2. CO-TRAINING

The co-training algorithm proposed by Blum and Mitchell [1] is given in Table 1. The co-training paradigm assumes that the features can be split into two disjoint sets (views) V1 and V2. For example, consider the task of classifying e-mails as spam and non-spam. Each e-mail can be described by the words in the subject (V1) and the words in the body (V2). The two classifiers C1 and C2 are trained using one view of the small labelled data L which results in two weak classifiers. Then each of them assigns labels to all unlabelled examples U, selects the most confidently predicted and moves them from the unlabelled to the labelled set. Both classifiers are re-trained on the enlarged labelled set. The loop is repeated for a pre-defined number of iterations. When training is completed, the label of new instances is predicted by multiplying the probabilities output by C1 and C2 and choosing the most probable class.

In essence, the two algorithms train each other. For example, suppose that C1 can confidently and correctly predict the class of an unlabelled example in V1, for which C2 is unsure in V2. By adding this example to the training set, C2 extends its knowledge and is able to learn better in future. For our example, suppose that C1 confidently classifies e-mails with the word "free" in the subject as spams, while C2 is not sure based on the information in the body. By transferring this example in the labelled set, C2 will learn that the words in the body indicate class "spam".

Given:
- a small set **L** of labelled examples
- a large set **U** of unlabelled examples
- two feature sets (views) $\mathbf{V_1}$ and $\mathbf{V_2}$ describing the examples

Training:
Loop for ***k*** iterations:
Learn classifier $\mathbf{C_1}$ from **L** based on $\mathbf{V_1}$
Learn classifier $\mathbf{C_2}$ from **L** based on $\mathbf{V_2}$
$\mathbf{C_1}$ labels examples from **U** based on $\mathbf{V_1}$ and chooses the most confidently predicted **p** positive and **n** negative examples $\mathbf{E_1}$
$\mathbf{C_2}$ labels examples from **U** based on $\mathbf{V_2}$ and chooses the most confidently predicted **p** positive and **n** negative examples $\mathbf{E_2}$
$\mathbf{E_1}$ and $\mathbf{E_2}$ are removed from **U** and added with their labels to **L**
End

Classification of new examples:
Multiply the probabilities that are output by C1 and C2

Table 1- Co-training algorithm

3. BASE CLASSIFIERS

As base classifiers for the co-training algorithm we chose Naïve Bayes (NB) and Random Forest (RF). NB is the classifier predominantly used in a co-training setting and we chose it as a benchmark. RF is a new, less popular but very efficient algorithm in terms of both predictive power and running time. To the best of our knowledge it hasn't been applied for image classification or in a co-training setting.

3.1. Random Forest

RF [5] is a recently introduced ensemble approach that combines decision trees [6]. Diverse trees, forming the RF, are generated by both altering the data set using bagging and selecting random input features. If n is the number of training examples and m is the number of features in the original training data, the training data for each of the t ensemble members is first generated by randomly selecting n instances from the training data with replacement. Then, for each data sample, a decision tree is grown. When growing a typical decision tree, splits on all available attributes for a given node are considered and the best one is selected based on performance indexes such as Information Gain or Gini [6]. In RF, only a small number k ($k<<m$) of randomly selected features, available at the node, are searched. Their number k is kept fixed but for each split a new random set of features of size k is selected. Each tree is fully grown and not pruned as opposite to the standard decision tree which is typically pruned. To classify a new example, it is propagated through all t trees and the decision is taken by a majority vote.

The predictive power of RF depends on the strength of the individual trees and their correlation with each other. A tree with high strength has a low classification error. Ideally we would like the trees to be less correlated and highly accurate. As the trees become less accurate or correlated, the RF's performance decays. The low level of correlation is achieved by using bagging and random feature selection which inject randomness and generate dissimilar, and thus, low-correlated, trees. The strength and correlation also depend on the number of features k. As k increases, both the correlation among the trees and their accuracy tend to increase. As a trade-off, the value of k is typically set to $k = \log_2 m + 1$ [5].

RF has been shown to run much faster and give comparable accuracy results to the highly successful AdaBoost ensemble algorithm [5]. It was also proved that RF does not overfit. The ability to run efficiently on large data sets and produce accurate results makes RF a very attractive algorithm for image classification. In our experiments we used RF of 10 trees. The number of random features was set using the heuristic discussed above, resulting in 7-9 features.

3.2. Naïve Bayes

Naïve Bayes (NB) [6] is a very popular, simple and highly effective Bayesian learner. It uses the training data to estimate the probability that an example belongs to a particular class. It assumes that feature values are independent given the class. Although this assumption clearly has no basis in most learning situations, NB can produce very good results.

3.3. Using Different Base Classifiers

The standard co-training algorithm involves the use of two classifiers of the same type, e.g. two NBs. Different classification algorithms have different inductive bias, i.e. they use different representations for their hypothesis, search differently the space of all possible hypotheses and avoid overfitting in different ways [6]. There is a group of ensembles that exploit this diversity between algorithms by combining different classifiers to create an effective ensemble. The same idea can be used in a co-training setting. Two different algorithms as base classifiers may be more capable of helping each other by focusing on different aspects of the data, and not making the same mistakes. This also may help to improve the task coverage problem [4]. The classifiers label most confidently examples from the task space most familiar to them. Different types of classifiers are likely to be familiar with different parts of the task and, hence, to select more useful examples.

This idea is similar to the work of Goldman and Zhou [7] who also used different classifiers but one of the classifiers is labelling examples for the other while in our case the labelled examples are used by both classifiers. Their algorithm

also requires that the classifiers partition the example space into a set of equivalence classes and uses confidence intervals to decide when one of the classifiers should label for the other, and also to combine the decisions of the two classifiers. We stick to the standard co-training setting and investigate if the use of two different classifiers is beneficial.

4. DATASETS

4.1. Task 1: Indoor-Outdoor Scene Classification

We used the ISB dataset [8]. It consists of 153 photos taken during and after the construction of the Information Services building of the University of Otago. It is a diverse data containing images of the construction site, the completed building with outdoor background, indoor scenes of close-ups of library users, and close-ups of indoor and outdoor architectural structures. As Figure 1 shows, some of the indoor and outdoor images have similar colour and components, especially architectural elements, which makes the classification difficult. The images were manually labelled as *indoors* and *outdoors*, resulting in 60 indoors and 93 outdoors images.

a) indoors

b) outdoors

Figure 1- Sample images from the ISB dataset

The first view for co-training was formed from the colour histogram features, the second one from the luminance edge histogram features. The colour histogram was computed in the LUV space, quantizing each channel into 5 bins which resulted in a 125 dimensional feature vector. For the edge histogram we used the MPEG-7 edge histogram descriptor [9]. Edge filters are first applied to detect the edges of the 2x2 pixel blocks (vertical, horizontal, 45°, 135° diagonal edges, non-directed edges and no edges). Grouping the edge information of all blocks generates an edge histogram with 6 bins. An image is partitioned into 4x4 sub-images, each generating a sub-image histogram. Concatenating these histograms results in a 96-dimensional global edge histogram.

4.2. Task 2: Animals-Sports Scene Classification

We used a subset of the Swedish University Network (SUNET) dataset [10]. We chose two categories: *animals* and *sports*. There are 511 images in *animals* and 415 in *sports*. Figure 2 shows some sample images. The images are very different in terms of objects and background. The first view for co-training was formed using the MPEG- 7 colour layout descriptor [9] which captures the spatial colour distribution in an image. This standard descriptor is computed by obtaining the representative colours in the YCrCb space on an 8x8 grid, DCT transforming and quantizing them into integers. To keep better precision, we used the raw average colours in YCrCb without the DCT transform and quantization. This resulted in a feature vector of 64x3=192 dimensions. The second view was formed using the global edge histogram (96 features) as described in task 1.

a) animals

b) sports

Figure 2- Sample images from the SUNET dataset

5. EXPERIMENTAL SETUP

5.1. Evaluation methodology

For the evaluation of the co-training results we used a procedure that resembles 10-fold cross validation. The 10-fold cross validation uses 90% of the data for training and 10% for testing. The co-training, on the other hand, uses only a small number of labelled and unlabelled training examples. Therefore, if 10-fold cross validation is applied in a co-training setting, many examples will not be used neither for training nor for testing. A better utilization of the available data is to increase the size of the test set which will improve the evaluation of the classifier without significantly reducing the quality of the classifier.

We generate 10 disjoint stratified folds. Each time 40% of the data is used for testing and the remaining 60% for training. The required amount of initially labelled examples are randomly selected from the first fold of the training data, and the remaining examples from this fold and the next 5 folds are used as unlabelled examples. The experiments are repeated 10 times and the results averaged, each time using

different fold to select the labelled examples, and creating different unlabeled and test sets by sliding 1 fold to the right.

5.2. Co-training Parameters

In task 1, the labelled set consisted of 4 indoors and 4 outdoors examples, and 1 newly labelled example from each class was added at a co-training iteration. For the bigger dataset in task 2, the labelled set consisted of 5 examples from each class, and 5 examples per class were added. For both tasks the number of co-training iterations was set to 20.

The number of examples from each class added at a co-training iteration is typically chosen following the class distribution. In our case it is balanced for both tasks (50-60%), so we add equal number of examples from the two classes.

6. RESULTS AND DISCUSSION

6.1. View strength

We first evaluate the strength of the individual feature sets (views). A view is said to be sufficiently strong if when taken on its own, is sufficient for accurate classification [2]. This property can be measured as the accuracy of the classifier trained on the fully labelled version of the training set for each data view. In our experiments we report the results on the test set using 10-fold cross validation.

	V1 (colour)	V2 (edge)
Task 1: indoors vs outdoors		
RF	81.2	84.6
NB	80.0	90.3
Task 2: animals vs sports		
RF	77.4	80.0
NB	73.6	80.5

Table 2- Classification accuracy [%] on the full labelled version of the training set for each data view

The strength of the two views is shown in the first two columns of Table 2. The baseline accuracy[1] is determined by the class distribution and it is 60.8% for task 1 and 55.2% for task 2. Thus, the two views on both tasks are reasonably strong. For RF the two views are of similar strength, while for NB V2 is stronger than V1 with 10-17%.

6.2. Co-training evaluation

The co-training results are given in Table 3. The column *it0* shows the accuracy of the combined classifier trained on the initial set of 10 labelled examples before the co-training (iteration 0). The column *it20* shows the accuracy at the end of co-training, i.e. after iteration 20, where the number of training examples is 48 for task 1 (8 initially labelled +20x2 self-labelled) and 210 for task 2 (10 initially labelled +20x10 self-labelled). The column *increase* presents the difference between iteration 20 and 0; thus, positive numbers indicate improvement over the base classifier trained on the initial set of labelled examples, and negative numbers indicate decline in performance. The column *gap* indicates the difference between the goal accuracy and *it20*'s accuracy. As a goal accuracy we consider the accuracy of a classifier trained on the labelled version of all training data (both views), using 10-fold cross validation (i.e. trained on 138 examples with 221 features in task 1, and 827 examples with 288 features in task 2). Figures 3 and 4 present the co-training learning curves, together with the goal accuracies. For the mixed classifier RF-NB, the goal accuracy is the highest of NB goal and RF goal.

	it0	it20	increase (it20-it0)	gap (goal-it20)
Task 1: indoors vs outdoors				
RF-RF	79.1	78.4	-0.7	11.9
NB-NB	82.4	87.7	5.3	1.5
RF-NB	82.9	84.3	1.4	7.4
Task 2: animals vs sports				
RF-RF	70.0	78.2	8.2	3
NB-NB	73.5	76.4	2.9	1.2
RF-NB	76.9	79.7	2.8	1.5

Table 3 - Co-training results: accuracy [%] of combined classifier

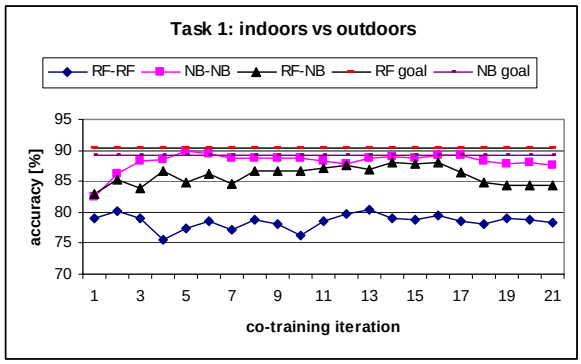

Figure 3 - Co-training on task 1

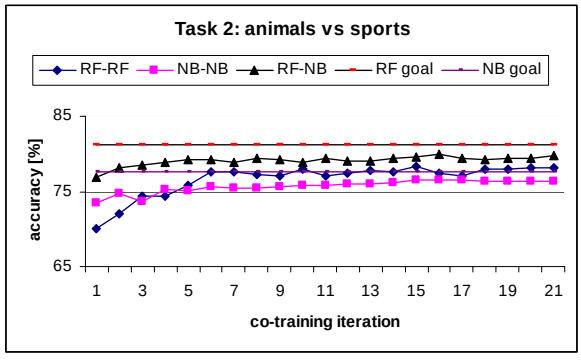

Figure 4 - Co-training on task 2

On task 1 NB-NB was the most successful classifier: it improved over the initial classifier achieving the highest accuracy value and smallest gap. The mixed classifier RF-NB came second and also benefited from co-training. RF-RF achieved an improvement over the initial classifier in itera-

[1] The so called ZeroR accuracy is calculated by assigning each test example to the majority class in the training data.

tion 12 but was not able to sustain it and by iteration 20 decreased the performance of the initial classifier with 0.7%.

On task 2, all classifiers were successful in a co-training environment, improving the accuracy of the initial classifier. The mixed classifier obtained the highest accuracy value, followed by RF-RF and NB-NB. The smallest gap was obtained by NB-NB and the highest improvement by RF-RF.

Overall, NB-NB was the most successful classifier in reducing the gap. Co-training with NB using 8-10 labelled examples and trained for 20 iterations achieved an accuracy rate of only 1.2-1.5% lower that NB trained on the labelled version of the training set (138 examples in task 1 and 827 in task 2).

The mixed classifier RF-NB was overall the best classifier in terms of accuracy value, coming second in task 1 and first in task 2. Recall that on task 2 both RF and NB performed very well individually, labelling correctly the most confident examples. Given this, the mixed co-training classifier was able to additionally benefit from the different inductive biases of the two algorithms and improve performance. On task 1 there was a big difference in the individual performance of NB and RF, and RF-NB was able to improve over the weaker of them but not to outperform the stronger.

To summarise, the results show that co-training can be successfully used for image classification. However, it is sensitive to the base classifier used, which confirms the observation of Kiritchenko and Matwin [3]. On the task of e-mail filing into folders they found that support vector machines benefited from co-training while the performance deteriorated when using NB. Despite this sensitivity to the classifiers used, our results show that co-training can benefit from using different base classifiers. When the two base classifiers worked well in a co-training setting, the mixed classifier outperformed both of them. When one of the base classifiers worked well but the other did not, the mixed classifier outperformed the unsuccessful base classifier.

In a practical application of co-training there are important questions that need to be answered: Given a small set of labelled examples, how do we know if co-training will be successful? How do we select the number of examples to be transferred to the labelled set at each iteration? How long should we run the co-training algorithm before the labelling accuracy deteriorates? Which is the best choice for base classifiers? All these are open research questions. The good news is that we may be able to learn these parameters in image classification tasks. In image classification very often we use the *same* feature sets to solve *different* problems; even in this study we use the same feature sets (colour and edge) for two different image classification tasks (*indoor*s vs *outdoors* and *animals* vs *sports*). Given enough examples of successful and unsuccessful co-training, we can learn to predict if co-training with these views will be useful in a new task, and what the parameters should be. For example, as meta-features one can use the accuracy of the classifier trained on the small labelled set, the agreement between these classifiers on the unlabelled data, dataset characteristics such as number and type of attributes, and more complex statistical measures. Thus, the task can be cast as meta-learning. Muslea et al. [11] described an approach that attempts to learn if two views are sufficiently compatible in multi-view learning for the tasks of wrapper induction and text classification. In the area of algorithm selection, we have developed an approach which, for a given task, selects the best classifier from a set of classifiers, given their previous performance on other problems [12]. We plan to extend these approaches to predict if a task is suitable for co-training, and what the co-training parameters should be.

7. CONCLUSIONS

We present a case study of co-training in the area of image classification. The results show that co-training with NB, using 8-10 labelled examples and trained for 20 iterations, is able to obtain accuracy only 1.2-1.5% lower that NB trained on the full labelled version of the training set (138 examples in task 1 and 827 examples in task 2). RF was shown to benefit from co-training in one of the tasks and slightly to decrease the performance of the initial classifier in the other task. Thus, co-training is sensitive to the choice of base classifier. The proposed modification of the co-training algorithm, motivated by the different inductive basis of classifiers, was shown to be a promising approach and needs further investigation. Another avenue for future work is to apply meta-learning to learn to predict if co-training is suitable for a new task, and what its parameters should be. Image classification is particularly suitable for this task as there are different tasks characterised by the same views.

REFERENCES

[1] A. Blum and T. Mitchell, " Combining Labeled and Unlabeled Data with Co-Training," Workshop Comp. Learning Theory, 1998.
[2] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training, ," Int. Conf. Inf.& Knowl. Manag., 2000.
[3] S. Kiritchenko and S. Matwin, "Email Classification with Co-Training," CASCON, 2001.
[4] D. Pierce and C. Cardie, "Limitations of co-training for natural language learning from large datasets," Empir.Methods NLP, 2001.
[5] L. Breiman, " Random Forests," *Machine Learn.g*, vol. 45, pp. 5-32, 2001.
[6] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*: Morgan Kaufmann, 2005.
[7] S. Goldman and Y. Zhou, "Enhancing Supervised Learning with Unlabelled Data," 17th Int. Conf. Machine Learning (ICML), 2000.
[8] D. Deng and J. Zhang, "Combining Multiple Precision-Boosted Classifiers for Indoor-Outdoor Scene Classification," ICITA, 2005.
[9] B. Manjunath, J. Ohm, & V. Vinod, "Color and Texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 703-715, 2001.
[10] "ftp://ftp.sunet.se/pub/pictures/."
[11] I. Muslea, S. Minton, C. Knoblock, "Adaptive View Validation A First Step towards Automatic View Detection," ICML, 2002.
[12] D. Ler, I. Koprinska, and S. Chawla, "Utilising Regression-Based Landmarkers within a Meta-Learning Framework for Algorithm Selection," Workshop on Meta-Learning, ICML, 2005.