Validating Co-Training Models for Web Image Classification

Dell Zhang[1,2] and Wee Sun Lee[1,2]
[1] Singapore-MIT Alliance, [2] Department of Computer Science, National University of Singapore

***Abstract*— Co-training is a semi-supervised learning method that is designed to take advantage of the redundancy that is present when the object to be identified has multiple descriptions. Co-training is known to work well when the multiple descriptions are conditionally independent given the class of the object. The presence of multiple descriptions of objects in the form of text, images, audio and video in multimedia applications appears to provide redundancy in the form that may be suitable for co-training. In this paper, we investigate the suitability of utilizing text and image data from the Web for co-training. We perform measurements to find indications of conditional independence in the texts and images obtained from the Web. Our measurements suggest that conditional independence is likely to be present in the data. Our experiments, within a relevance feedback framework to test whether a method that exploits the conditional independence outperforms methods that do not, also indicate that better performance can indeed be obtained by designing algorithms that exploit this form of the redundancy when it is present.**

***Index Terms*— Co-Training, Machine Learning, Multimedia Data Mining, Semi-Supervised Learning.**

I. Introduction

MULTIMEDIA applications are unique in providing many different descriptions (such as text, images, audio and video) of a single object or event. Quite often, these combined descriptions have various amounts of redundancies. Consequently information processing methods that take advantage of these redundancies may be able to outperform methods that do not.

Co-training [1] is a semi-supervised learning method that takes advantage of a particular form of redundancy in data to effectively learn the target function based on a few labeled and many unlabeled examples. In co-training models, it is assumed that there are (at least) two distinct views of an object, each of which contains enough information to identify the object. Furthermore, the two descriptions are conditionally independent, given the identity of the object. For example, the co-training assumptions imply that if we have a text description and an image of George Bush, then the words and pixels are individually enough to identify George Bush, and furthermore, the distribution of the pixels is independent of the words once we know that the object described is George Bush, and vice versa.

Under the assumption that each view contains enough information for identifying the object, an (unlabeled) example of the object with two views provides a link between descriptions of the two views. Under the conditional independence assumption, the probability of a description in the first view being linked to a description in the second view is just the probability of the description appearing in the second view. Hence, with a large enough sampling of unlabeled examples, most of the commonly occurring descriptions in the two views will be connected together allowing it to be easily learned.

With the explosive growth of the Web, an immense amount of multimedia information is becoming freely available online. With the prevalence of multiple descriptions of the same objects and events, it is desirable to develop methods that can exploit the available redundancy. In this paper, we provide a preliminary study of the possibility of exploiting the redundancies present in text and images from the same webpage. We use a simple setup of using relevance feedback to disambiguate ambiguous queries for images. We collected images and webpages for five ambiguous queries on a search engine. For each of these queries, we use two different unambiguous target classes that need to be disambiguated from the other images in order to form a total of ten relevance feedback tasks.

We first perform some measurements on the data in order to find evidence of conditional independence, which would suggest the usefulness of co-training methods for the task. We show some evidence that the conditional independence assumption is indeed reasonable. This is also supported by the results of the relevance feedback experiments. We show that the co-training method that exploits the conditional independence outperforms other methods that do not. To test whether conditional independence is the reason for co-training's superior performance, we construct two artificial views of each example by unnaturally splitting of features in the following way: one view is composed of the 1st half of the

Dell Zhang is with the Computer Science Program in Singapore-MIT Alliance, National University of Singapore, Singapore 117543 (phone: 65-6874-4251; fax: 65-6779-4580; e-mail: dell.z@ieee.org).

Wee Sun Lee is with the Computer Science Program in Singapore-MIT Alliance, and Department of Computer Science in National University of Singapore, Singapore 117543 (e-mail: leews@comp.nus.edu.sg).

image content features and the 2nd half of the associated text features while the other view is composed of the 2nd half of the image content features and the 1st half of the associated text features. Our measure of conditional independence dropped. Similarly, classification performance dropped, supporting the hypothesis that co-training works better on the tasks because of conditional independence.

The rest of this paper is organized as follows. In section II, we review the co-training method and describe the embedded classifiers used. In section III, we propose the quantitative measurements for estimating whether the domain is suitable for co-training. In section IV, we present the experiments within a relevance feedback framework. In section V, we survey the related work. In section VI, we give the conclusion.

II. CO-TRAINING

Co-training [1] is a semi-supervised learning method that takes advantage of a particular form of redundancy in data to effectively learn the target function based on a few labeled and many unlabeled examples.

For co-training to perform well, there should be a natural way to split the features into disjoint subsets called views, such that the target function on the views is "compatible" and instances from the views are "uncorrelated". For example, in a domain with two views **V1** and **V2**, any example $\mathbf{x}$ can be seen as a triple $\langle \mathbf{x}1, \mathbf{x}2, y \rangle$, where $\mathbf{x}1$ and $\mathbf{x}2$ are its descriptions in the two views V1 and V2 respectively, and y is its label. The "compatible" assumption means that the target concept $f = (f_1, f_2)$, where the domain of f_1 is V1 and the domain of f_2 is V2, satisfies $f_1(\mathbf{x}1) = f_2(\mathbf{x}2)$ for all instances $\langle \mathbf{x}1, \mathbf{x}2 \rangle$ drawn from the distribution of instances. Given this assumption, the unlabeled examples can be used to eliminate functions that are not "compatible", hence yielding information that can help to learn the target function, The "uncorrelated" assumption means that given the label of any example, its descriptions in different views are conditionally independent. This assumption ensures that the labels propagate throughout the domain through the unlabeled examples instead of being confined to part of the domain.

There exist two kinds of features for a Web image, its pixel content and its associated text, which are possibly redundant. We can often tell what an image is about by looking at the image itself or looking at only its associated text. That is to say, we can use the content features as one view and the text features as another. Intuitively these two views satisfy the "compatible" and "uncorrelated" requirements. Such a natural split of features makes Web image classification a good candidate for the application of co-training.

The co-training algorithm shown in Fig. 1 is adapted from [1]. The algorithm iteratively generates more training examples for classifiers in both views with the aim of finally generating a pair of classifiers that agree on both the labeled and the unlabeled examples.

input:
- two feature sets V_1 and V_2
- a set of labeled examples L
- a set of unlabeled examples U
- two parameters p and n

while there exist examples in U **do**
  C_1 teaches C_2:
  train classifier C_1 based on the V_1 portion of L
  classify all examples in U using C_1
  remove p positive and n negative examples from U on which C_1 makes the most confident predictions, and then add them with their predicted labels to L
  C_2 teaches C_1:
  train classifier C_2 based on the V_2 portion of L
  classify all examples in U using C_2
  remove p positive and n negative examples from U on which C_2 makes the most confident predictions, and then add them with their predicted labels to L
end while
train a classifier C^* based on the $V_1 \bigcup V_2$ portion of the expanded set of labeled examples L
classify all originally unlabelled examples using C^*
output:
- the class labels for all originally unlabelled examples

Fig. 1. Outline of the adapted Co-Training algorithm used in this paper.

A. Support Vector Machines

Support Vector Machine (SVM) [2] has good theoretical properties as a classifier and has been shown to perform well in many practical domains. SVM is well suited for classifying the images on the Web using image content or associated text features because it can automatically avoid the pitfalls of very high dimensional representations. In this paper, we use SVMs as the embedded classifiers of co-training.

SVM is essentially a linear function of the form $f(x) = \langle \mathbf{w} \bullet \mathbf{x} \rangle + b$, where $\langle \mathbf{w} \bullet \mathbf{x} \rangle$ is the inner product between the weight vector $\mathbf{w}$ and the input vector $\mathbf{x}$. The SVM can be used as a classifier by setting the class to 1 if $f(x) > 0$ and to -1 otherwise. The main idea of SVM is to select a hyper-plane that separates the positive and negative examples while maximizing the minimum margin, where the margin for example $\mathbf{x}_i$ is $y_i f(\mathbf{x}_i)$ and $y_i \in \{-1, 1\}$ is the target output. This corresponds to minimizing $\langle \mathbf{w} \bullet \mathbf{w} \rangle$ subject to $y_i(\langle \mathbf{w} \bullet \mathbf{x}_i \rangle + b) \geq 1$ for all i. Large margin classifiers are known to have good generalization properties (see e.g. [3]).

To deal with cases where there may be no separating hyper-plane, the soft margin SVM has been proposed. The soft margin SVM minimizes $\langle \mathbf{w} \bullet \mathbf{w} \rangle + C\left(J \sum_{i:y_i=+1} \xi_i + \sum_{i:y_i=-1} \xi_i \right)$

subject to $y_i(\langle \mathbf{w} \bullet \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$ and $\xi_i > 0$ for all i, where C is a parameter that controls the amount of training errors allowed, and J is a parameter that weights the training errors on positive examples over those on negative examples.

The sign of the SVM output for a test example indicates its predicted label. The magnitude of the SVM output for a test example can be geometrically regarded as the distance from that test example to the classifying hyperplane (decision boundary), therefore it can be used to indicate the degree of confidence we have on this classification.

III. Estimating Approximability and Uncorrelatedness

The co-training algorithm tries to get the two functions on both views to agree on their classifications. For it to be successful, a good approximation to a compatible target function should exist in the function classes used to train on the two views. This depends both on the features available in the two views as well as on the function classes used. Another requirement is that the two views are uncorrelated given the label. To find out whether the domain of Web images is suitable for co-training, we estimate the approximability and uncorrelatedness of the data. The measurements are done to help explain why co-training works (or does not work).

A. Approximability

To see whether or not a view (a set of features) is sufficient for learning, we use the leave-one-out estimation of *F*-score as computed for SVM in [4]. The *F*-score [5] is defined as the harmonic average of precision (*p*) and recall (*r*), $F = \frac{2pr}{p+r}$, where precision is the proportion of correctly predicted positive examples among all predicted positive examples, and recall is the proportion of correctly predicted positive examples among all true positive examples.

We measure the "approximability" of two views V1 and V2 by M_A, which is defined as the harmonic average of F_{V1} and F_{V2}, $M_A = \frac{2F_{V1}F_{V2}}{F_{V1}+F_{V2}}$ where F_{V1} and F_{V2} are the leave-one-out estimations of *F*-score in V1 and V2 respectively.

B. Uncorrelatedness

The feature space is usually high-dimensional, making it difficult to directly apply the definition of conditional independence for measuring "uncorrelatedness". In [6], the "uncorrelatedness" assumption of two views was checked by calculating the sum of pair-wise conditional mutual information for all pairs of features in different views.

We take a different approach by calculating the correlation between the real-valued outputs of the two SVMs trained on all labeled examples in the two views respectively. If the correlation between SVM outputs is high, we can confidently say that the two views are not conditionally independent. Otherwise we feel more confident that the two views are conditionally independent even though we are unable to ascertain that they are.

We measure the "uncorrelatedness" of two views V1 and V2 by M_U, which is defined as $M_U = \frac{(1-|R_P|)+(1-|R_N|)}{2}$, where R_P and R_N are the conditional correlation coefficients of $f_1(\mathbf{x}1_i)$ and $f_2(\mathbf{x}2_i)$ given that the label y_i is positive and negative respectively.

The standard statistical correlation coefficient R of a set of n data points (u_i, v_i) is defined as follows:

$$R^2 = \frac{SS_{uv}^{\ 2}}{SS_{uu}SS_{vv}}, \text{ where}$$

$$SS_{uv} = \sum (u_i - \overline{u})(v_i - \overline{v}),$$

$$SS_{uu} = \sum (u_i - \overline{u})^2,\ SS_{vv} = \sum (v_i - \overline{v})^2,$$

$$\overline{u} = \frac{\sum u_i}{n},\ \overline{v} = \frac{\sum v_i}{n}.$$

The coefficient R varies between -1 and 1 with 0 indicating that the two variables u_i and v_i are totally uncorrelated and 1 or -1 indicating that they are totally correlated.

C. Combined Measure

To give an indication of the suitability of the domain for co-training, we define the multi-view quality measure M_{MVQ} as the harmonic average of the approximability and uncorrelatedness measures, $M_{MVQ} = \frac{2M_AM_U}{M_A+M_U}$.

Harmonic average in the above definitions is because the harmonic average of two values has the following property: it is high when both of these two values are high, and if these two values are radically different it is dominated by the smaller one.

IV. Experiments

We conducted experiments on real-world Web image data in the relevance feedback framework to evaluate the effectiveness of our proposed approach.

A. Relevance Feedback

The performance of image retrieval systems is often limited by the gap between low-level features and high-level semantic concepts. To address this problem, relevance feedback [5] techniques can be applied to learn user's intentions and boost the system's performance. Basically it is to ask the user to give some feedbacks on the returned results and try to refine the retrieval function based on these feedbacks.

From the perspective of machine learning, relevance feedback can be re-phrased as a classification problem: the

retrieval system trains a binary classifier based on labeled examples provided by user's relevance feedback, then uses the learned classifier to classify the unlabeled examples into two classes: relevant and irrelevant.

A typical problem with relevance feedback is the relatively small number of training examples. The system can only present the user a few images to be labeled as relevant or irrelevant. This suggests that it is beneficial to utilize semi-supervised learning algorithms for relevance feedback.

B. Data

Google (http://www.google.com/) is considered the favorite search engine of general users. It provides image search function (http://images.google.com/).

TABLE I
DATA FOR WEB IMAGE CLASSIFICATION EXPERIMENTS IN RELEVANCE FEEDBACK FRAMEWORK, n IS THE NUMBER OF ALL IMAGES AND n_P MEANS THE NUMBER OF POSITIVE IMAGES

Query	*Target*	n_P	n
apache	helicopter	152	704
apache	landscape	61	704
apple	computer	27	288
apple	fruit	44	288
jaguar	animal	213	726
jaguar	car	191	726
Madonna	saint	434	835
Madonna	singer	214	835
Venus	painting	43	773
Venus	planet	156	773

We submitted 5 ambiguous queries to the Google image search engine and grabbed its search results as data for experiments. Only the full-color images would be taken into account, and all out-dated images which are no longer available online would be removed. For each query, we select 2 of its major semantic interpretations as target concepts for retrieval, thus we pose $5\times 2=10$ relevance feedback problems as shown in Table I. The images corresponding to the target concept are labeled as positive and others as negative. These problems are very difficult for learning algorithms because the positive images are minorities in the extremely unbalanced data. This situation is quite common in retrieval applications.

C. Features

1) Image Content Features

In this paper, we choose to use color histograms as image content features, because of reasonable performance that can be obtained using color histograms in spite of its extreme simplicity [7].

Each particular color can be described as a position in a three dimensional color space. The color space HSV (Hue Saturation Value) is used instead of RGB (Red Green Blue), because the former de-correlates the color components (HS) from the luminance component (V) and is argued to be cognitively more plausible. Every color component is divided evenly into 16 bins, so that the dimension of the feature space is $16^3 = 4096$ [7]. Then an image is considered as a color histogram $\mathbf{h} = (h_1, h_2, ..., h_m)$, where h_i encodes the fraction of pixels of the ith color. An obvious advantage of this representation is that it is invariant with respect to many operations like scale, translation and rotation. To make the image content features more linearly separable, a nonlinear map is applied to the color histograms $\mathbf{x} = \Phi(\mathbf{h}) = (h_1^a, h_2^a, ..., h_m^a)$, where an appropriate value for a is 0.25 [7]. Finally all content feature vectors are normalized to have unit length.

Although this feature extraction technique is a very simple low-level method, it has shown good results in practice for image classification [7].

The content features for each Web image are extracted from not the original image but its corresponding thumbnail-image in the search results, due to efficiency consideration.

2) Associated Text Features

In addition to its content, a Web image can also be described by its occurring context, i.e., the text of the webpage which contains it. One can often tell what an image is about by only reading its associated text.

The most commonly used feature extraction technique for text is to treat a document as a bag-of-words [4]. For each document d in a collection of documents D, its bag-of-words is first pre-processed by removal of stop-words and stemming. Then it is represented as a feature vector $\mathbf{x} = (x_1, x_2, ..., x_m)$, where x_i indicates the importance weight of term w_i (the i-th distinct word occurred in D). Following the TF*IDF weighting scheme, we set the value of x_i to the product of the term frequency $TF(w_i, d)$ and the inverse document frequency $IDF(w_i)$, i.e., $TF(w_i, d) * IDF(w_i)$. The term frequency $TF(w_i, d)$ means the number of times w_i occurred in d. The inverse document frequency is defined as $IDF(w_i) = \log\left(\frac{|D|}{DF(w_i)}\right)$, where $|D|$ is the total number of documents in D, and $DF(w_i)$ is the number of documents in which w_i occurred. Finally all feature vectors are normalized to have unit length.

3) Both Features

We can choose to represent a Web image using both content features and text features by simply pooling them together. In this case, a combined feature vector with unit length is constructed for each Web image as $\mathbf{x} = (\mathbf{x}1, \mathbf{x}2)/\sqrt{2}$, where $\mathbf{x}1$ and $\mathbf{x}2$ are the normalized content and text feature vectors respectively.

D. Approximability and Uncorrelatedness

The approximability and uncorrelatedness measurements for the image and text views of the experimental data are listed in Table II.

TABLE II
QUALITY ESTIMATIONS FOR THE CONTENT AND TEXT VIEWS OF THE WEB IMAGE DATA

Query	*Target*	M_A	M_U	M_{MVQ}
apache	helicopter	0.63	0.96	0.76
apache	landscape	0.33	0.98	0.50
apple	computer	0.53	0.89	0.67
apple	fruit	0.43	0.96	0.60
jaguar	animal	0.69	0.93	0.79
jaguar	car	0.65	0.92	0.76
Madonna	saint	0.75	0.92	0.82
Madonna	singer	0.66	0.91	0.77
Venus	painting	0.38	0.87	0.53
Venus	planet	0.66	0.88	0.76
Average		0.57	0.92	0.70

In addition, to confirm our speculations that the content and text views formed by a natural split of features is ideal for co-training, we also design two artificial views formed by an unnatural split of features in the following way: the artificial "*content*" view is composed of the 1st half of the content features and the 2nd half of the text features; and the artificial "*text*" view is composed of the 2nd half of the content features and the 1st half of the text features. The quality estimations for the artificial views of the experimental data are listed in Table III.

TABLE III
QUALITY ESTIMATIONS FOR TWO VIEWS FORMED BY AN UNNATURAL SPLIT OF FEATURES OF THE WEB IMAGE DATA

Query	*Target*	M_A	M_U	M_{MVQ}
apache	helicopter	0.71	0.61	0.66
apache	landscape	0.45	0.78	0.57
apple	computer	0.60	0.80	0.68
apple	fruit	0.49	0.82	0.61
jaguar	animal	0.73	0.60	0.66
jaguar	car	0.71	0.57	0.63
Madonna	saint	0.78	0.46	0.58
Madonna	singer	0.73	0.68	0.70
Venus	painting	0.40	0.81	0.54
Venus	planet	0.78	0.62	0.69
Average		0.64	0.68	0.63

As can be seen from the tables, the uncorrelatedness measurements for the natural split of Web image features are high. Unfortunately, the approximability scores are not as good. The reason could be that the features are not powerful enough for very good classification using linear functions. The small number of training examples particularly positive examples also contribute to the low scores. For the artificial views formed by an unnatural split of features, the approximability scores are interestingly higher. However, as expected, the uncorreletedness scores are lower resulting in lower average overall score. The lower overall score suggests that the unnatural split may perform poorer when applying co-training even though its approximability score is higher. Indeed, this has been observed in our experiments.

Fig. 2 and 3 illustrate the plot of SVM outputs trained on all examples in two views with moderate uncorrelatedness (M_U=0.464), and high uncorrelatedness (M_U=0.916) respectively. Fig. 2 is generated using the artificial views from the 'Madonna-saint' experiment, which are formed by an unnatural split of features as mentioned before. Fig. 3 is generated using the content and text views from the 'Madonna-saint' experiment.

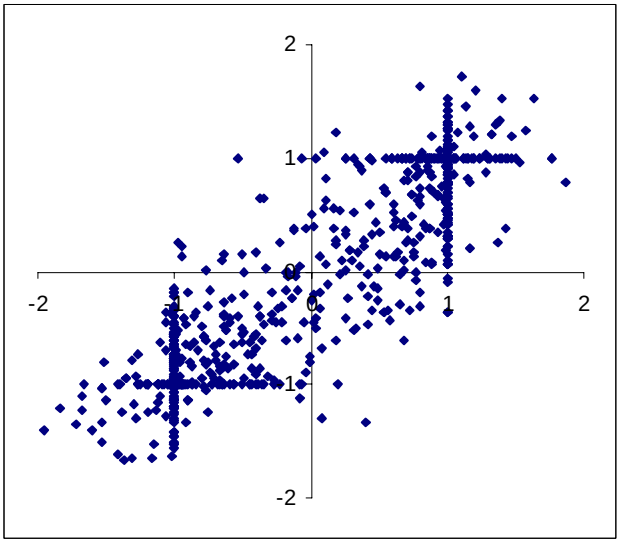

Fig. 2. Plot of SVM outputs in two views with moderate uncorrelatedness (M_U = 0.464).

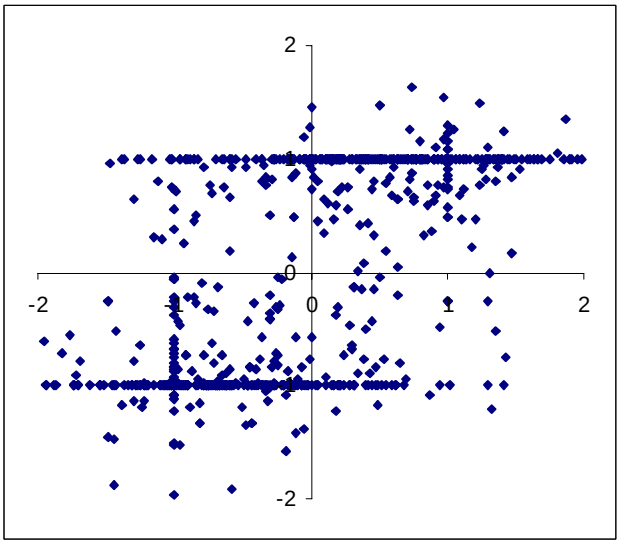

Fig. 3. Plot of SVM outputs in two views with high uncorrelatedness (M_U = 0.916).

E. Algorithms for Comparison

In this paper, we compare the proposed *co-training SVMs* approach with three other approaches that do not take advantage of the conditional independence, namely *supervised SVM*, *one-class SVM*, and *transductive SVM*.

The supervised SVM algorithm is just the regular SVM that is trained on labeled examples. It has been applied for relevance feedback in image retrieval systems [8].

The one-class SVM algorithm [9] is trained using only

positive labeled examples. It has been applied for relevance feedback in image retrieval systems [10], under the assumption that positive examples cluster in certain way, but negative examples usually do not cluster because they belong to several different classes.

The transductive SVM algorithm [11] is also a semi-supervised learning algorithm. It tries to assign labels to the unlabeled data in such a way that resulting classifier has large margin in both the labeled and unlabeled data. It is particularly beneficial in the situations where we do not care about good generalization, but rather good classification accuracy on a particular test set.

F. Settings

For each retrieval problem, we run the learning algorithm 10 times, using 20 randomly selected examples to simulate the user's relevance feedback at each time. We choose the number 20 because Google image search engine displays 20 images in its first search result page. The standard *F*-score [5] is used to evaluate the retrieval performance.

An efficient implementation of SVM, *SVMlight* (http://svmlight.joachims.org/), is employed throughout our experiments, except that the implementation of one-class SVM is from *libSVM* (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). We always choose the linear kernel as it has been shown that SVM with linear kernel works quite well for image classification based on color histogram features [7], and for text classification based on TF*IDF features [4]. The SVM parameter C is set as its default value, i.e., $C = 1$; the SVM parameter J is automatically adjusted to the value $J = n_N / n_P$, i.e., the ratio of the number of training positive examples to the number of training negative examples, in order to give equal importance to both positive and negative examples.

The co-training SVMs algorithm is designed to keep running while there are enough unlabeled examples. In every round of co-training, given that there are *p* positive and *n* negative examples in the 20 labeled examples provided by the user's relevance feedback, the SVM classifier in either content or text view fetches *p* positive and *n* negative most confidently labeled examples to the training set. That is to say, there are about 20+20=40 new examples added to the training set, while the proportion of positive or negative examples is forced to be consistent with the initial proportion in the user's relevance feedback. At last the enlarged training set is used to train three SVMs based on content, text, and both features respectively, as the final learned classifiers.

Since all the above algorithms are based on SVM, we think that the leave-one-out estimation of the supervised SVM trained on all training and test data with the labels of test examples known can be regarded as the performance upper-bound. Please note that *SVMlight* is able to compute leave-one-out estimations very efficiently using a clever algorithm that prunes away cross-validation folds that do not need to be explicitly executed [4]. We have used a faster approximate version of pruning (the options "-x 1" and "-o 1") in our experiments.

G. Results

The experiment results are summarized in Table IV, where the *F*-score of each learning algorithm for each retrieval problem averaged on 10 runs is shown.

One interesting phenomena revealed by the experiment results is that using "both" features is not necessarily better than just using the content or text features, i.e., simply pooling content and text features together may not work.

The supervised SVM algorithm works poorly due to the lack of enough training examples. The one-class SVM algorithm is even worse, because it totally neglects the valuable knowledge provided by the labeled negative examples. The performance of the supervised SVM algorithm is actually the starting point of the co-training SVMs algorithm. Comparing the performances of the SVM classifiers before and after co-training, we see that co-training really can boost the classification performances of its embedded classifiers.

The semi-supervised learning algorithms, including co-training SVMs and transductive SVM, appear to be beneficial when there are only a few labeled examples and lots of unlabeled examples as in the relevance feedback case. The co-training SVMs algorithm behaves better than the transductive SVM algorithm, which suggests that leveraging the uncorrelatedness of this problem is helpful. The "Unnatural Co-Training" column in Table IV records the performance of the co-training SVMs algorithm with the artificial views formed by an unnatural split of features as described in the previous section. Although the approximability of the unnatural split is better, the overall performance is poorer than the natural split, indicating that uncorrelatedness is indeed helpful for co-training in this case.

To better understand the behavior of the co-training SVMs approach, we plot its learning curves of one run for the 'Venus-planet' retrieval problem. At the end of every round, we train three SVMs using the ever growing training set based on content, text, and both features respectively, and test them using the initially unlabeled examples (i.e., the images outside of the user's relevance feedback), their performances changing along the co-training rounds are depicted in Fig. 4, 5, and 6. One can see that the performances of all these three SVMs keep increasing during co-training.

TABLE IV
PERFORMANCE OF MACHINE LEARNING ALGORITHMS FOR WEB IMAGE CLASSIFICATION

Query	*Target*	*Features*	*Upper-bound*	*Supervised SVM*	*One-Class SVM*	*Transductive SVM*	*Unnatural Co-Training*	*Co-Training SVMs*
apache	helicopter	content	0.56	0.31	0.02	0.47	0.46	0.52
apache	helicopter	text	0.73	0.06	0.00	0.52	0.46	0.52
apache	helicopter	both	0.79	0.14	0.00	0.49	0.46	0.52
apache	landscape	content	0.25	0.06	0.00	0.12	0.16	0.20
apache	landscape	text	0.48	0.03	0.01	0.30	0.17	0.19
apache	landscape	both	0.48	0.02	0.00	0.15	0.16	0.19
apple	computer	content	0.60	0.16	0.00	0.35	0.48	0.46
apple	computer	text	0.48	0.05	0.00	0.29	0.49	0.46
apple	computer	both	0.67	0.04	0.00	0.41	0.49	0.47
apple	fruit	content	0.57	0.24	0.00	0.39	0.41	0.46
apple	fruit	text	0.35	0.02	0.00	0.40	0.42	0.44
apple	fruit	both	0.51	0.06	0.00	0.47	0.42	0.45
jaguar	animal	content	0.64	0.49	0.11	0.54	0.56	0.63
jaguar	animal	text	0.75	0.10	0.00	0.61	0.55	0.63
jaguar	animal	both	0.78	0.33	0.00	0.55	0.56	0.63
jaguar	car	content	0.60	0.41	0.07	0.42	0.41	0.53
jaguar	car	text	0.71	0.06	0.00	0.55	0.41	0.52
jaguar	car	both	0.78	0.26	0.00	0.44	0.41	0.53
Madonna	saint	content	0.68	0.55	0.26	0.56	0.57	0.61
Madonna	saint	text	0.83	0.43	0.01	0.60	0.56	0.61
Madonna	saint	both	0.79	0.50	0.03	0.56	0.57	0.61
Madonna	singer	content	0.60	0.22	0.07	0.30	0.31	0.40
Madonna	singer	text	0.75	0.04	0.01	0.41	0.31	0.39
Madonna	singer	both	0.79	0.09	0.01	0.30	0.31	0.39
Venus	painting	content	0.51	0.21	0.00	0.25	0.23	0.34
Venus	painting	text	0.31	0.01	0.00	0.21	0.23	0.35
Venus	painting	both	0.52	0.04	0.00	0.32	0.23	0.36
Venus	planet	content	0.71	0.32	0.03	0.44	0.52	0.66
Venus	planet	text	0.62	0.14	0.00	0.46	0.52	0.57
Venus	planet	both	0.81	0.23	0.00	0.48	0.52	0.59
Average			0.62	0.19	0.02	0.41	0.41	0.47

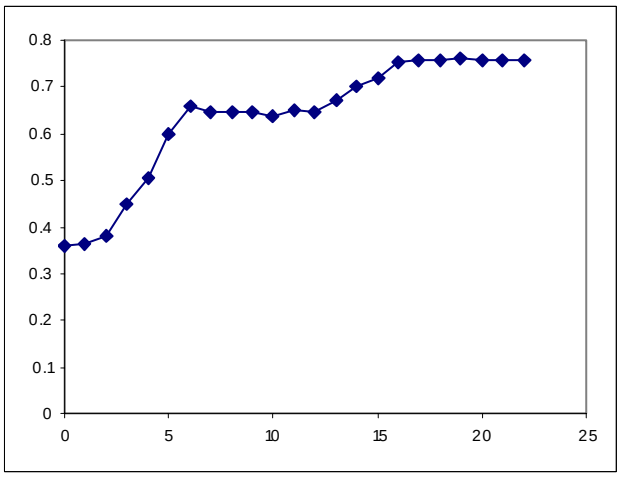

Fig. 4. Performance (F-score) of the SVM based on content features at Co-Training rounds.

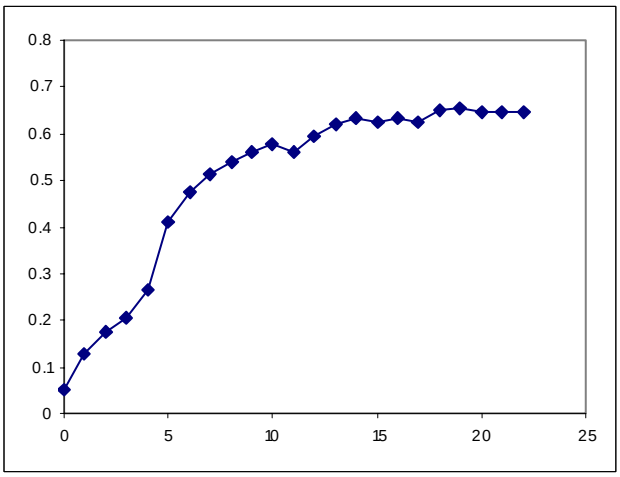

Fig. 5. Performance (F-score) of the SVM based on text features at Co-Training rounds.

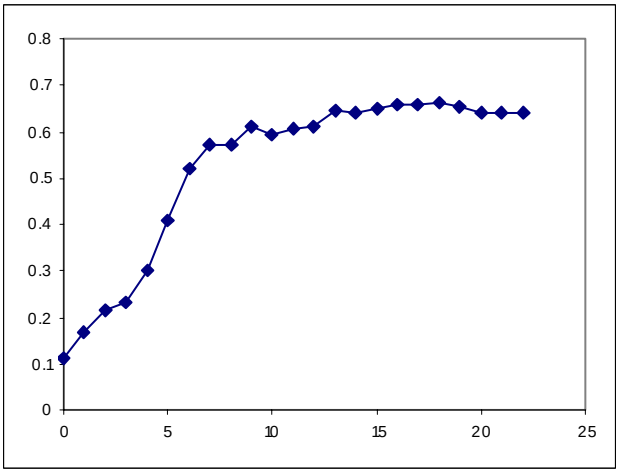

Fig. 6. Performance (F-score) of the SVM based on both features at Co-Training rounds.

V. Related Work

Several successful applications of co-training have emerged recently. In [1], co-training is applied to webpage classification, where the text of the webpage forms one view and the anchor text of links pointing to the same webpage forms another view. In [12], co-training is applied to named entity classification, where the spelling features of the named entity forms one view and the context of the named entity forms another view. In [13], co-training is applied to email classification, where the text of the email subject forms one view and the text of the email body forms another view.

The conditional independence property which is useful to co-training may not hold for many other real-world problems. Recent advances in co-training try to cope with this problem. In [6, 12, 14-16], refined co-training algorithms under relaxed assumptions are proposed. In [17], a meta-learning approach is proposed to discriminate between tasks for which the given two views are sufficiently/insufficiently compatible for co-training. It is interesting to combine co-training and active learning. In [18], humans are required to correct inaccurate labels made by co-training. In [19], the system asks users to explicitly label examples on which the two classifiers of co-training have different predictions.

Semi-supervised learning algorithms such as co-training and transductive SVM have gained more and more attention from the machine learning community in recent years. In [20], it is shown that Expectation Maximization (EM), a popular iterative statistical technique for maximum likelihood estimation in problems with incomplete data [21], can be employed to improve text classification using unlabeled data. In [15, 22], the graph mincut algorithm is applied to semi-supervised learning. In [23], it is shown that Latent Semantic Indexing (LSI) [24] can be employed to improve text classification using unlabeled data. In [25], a co-training like algorithm is proposed. In [26-28], special semi-supervised learning algorithms based on kernel classifiers especially SVMs are designed. In [29], a regularization approach to semi-supervised learning is proposed. In [6], it is empirically demonstrated that co-training outperforms EM even on tasks without natural split of features.

There has been a great increase of interest in Content-Based Image Retrieval (CBIR) after the well-known QBIC system from IBM [30] appeared. CBIR systems allow the user to find image visually similar to a given example image based on image content features, such as color, texture and shape. A recent article [31] reviewed more than 200 references in this ever changing filed. Obviously the image content features used in such CBIR systems may also be used in the co-training SVMs algorithm to improve on the color histogram used here.

VI. Conclusion

We have performed preliminary measurements that indicate the conditional independence which is useful for co-training is likely to exist in image and text views of the same concepts on the Web. We have also shown that co-training, which exploits the conditional independence that is present in the data, outperforms other methods that do not exploit conditional independence in a relevance feedback task on identifying objects with text and image descriptions.

Acknowledgment

We would like to thank Prof. Tomas Lozano-Perez and Prof. Leslie Kaelbling for useful discussion on this work.

References

[1] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*. Madison, WI, 1998, pp. 92-100.

[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press, 2000.

[3] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed: Springer-Verlag, 2000.

[4] T. Joachims, *Learning to Classify Text using Support Vector Machines*: Kluwer, 2002.

[5] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison-Wesley, 1999.

[6] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-training," in *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*. McLean, VA, 2000, pp. 86-93.

[7] O. Chapelle, P. Haffner, and V. N. Vapnik, "SVMs for Histogram Based Image Classification," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1055-1064, 1999.

[8] Q. Tian, P. Hong, and T. S. Huang, "Update Relevant Image Weights for Content-Based Image Retrieval Using Support Vector Machines," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2. New York, NY, 2000, pp. 1199-1202.

[9] B. Scholkopf, J. Platt, J. Shawe-Taylor, and A. J. Smola, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, pp. 1443-1471, 2001.

[10] Y. Chen, X. Zhou, and T. S. Huang, "One-class SVM for Learning in Image Retrieval," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1. Thessaloniki, Greece, 2001, pp. 34-37.

[11] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines," in *Proceedings of the 16th International Conference on Machine Learning (ICML)*. Bled, Slovenia, 1999, pp. 200-209.
[12] M. Collins and Y. Singer, "Unsupervised Models for Named Entity Classification," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP)*. College Park, MD, 1999, pp. 189-196.
[13] S. Kiritchenko and S. Matwin, "Email Classification with Co-Training," in *Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON)*. Toronto, Canada, 2001.
[14] S. P. Abney, "Bootstrapping," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia, PA, 2002, pp. 360-367.
[15] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*. Washington DC, USA, 2003, pp. 290-297.
[16] U. Brefeld and T. Scheffer, "Co-EM Support Vector Learning," in *Proceedings of the 21st International Conference on Machine Learning (ICML)*. Banff, Alberta, Canada, 2004.
[17] I. Muslea, S. Minton, and C. A. Knoblock, "Adaptive View Validation: A First Step Towards Automatic View Detection," in *Proceedings of the 19th International Conference on Machine Learning (ICML)*,. Sydney, Australia, 2002, pp. 443-450.
[18] D. Pierce and C. Cardie, "Limitations of Co-Training for Natural Language Learning from Large Datasets," in *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Pittsburgh, PA, 2001.
[19] I. Muslea, S. Minton, and C. A. Knoblock, "Active + Semi-Supervised Learning = Robust Multi-View Learning," in *Proceedings of the 19th International Conference on Machine Learning (ICML)*. Sydney, Australia, 2002, pp. 435-442.
[20] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, vol. 39, pp. 103-134, 2000.
[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1-38, 1977.
[22] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data using Graph Mincuts," in *Proceedings of the 18th International Conference on Machine Learning (ICML)*. Williamstown, MA, 2001, pp. 19-26.
[23] S. Zelikovitz and H. Hirsh, "Using LSI for Text Classification in the Presence of Background Text," in *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*. Atlanta, GA, 2001, pp. 113-118.
[24] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, vol. 41, pp. 391- 407, 1990.
[25] S. Goldman and Y. Zhou, "Enhancing Supervised Learning with Unlabeled Data," in *Proceedings of the 17th International Conference on Machine Learning (ICML)*. Standord, CA, 2000, pp. 327-334.
[26] K. P. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 11. Denver, CO, 1998, pp. 368-374.
[27] O. Chapelle, B. Scholkopf, and J. Weston, "Semi-Supervised Learning through Principal Directions Estimation," in *ICML Workshop, The Continuum from Labeled to Unlabeled Data in Machine Learning & Data Mining*, 2003.
[28] O. Chapelle, J. Weston, and B. Scholkopf, "Cluster Kernels for Semi-Supervised Learning," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15. Vancouver, Canada, 2003, pp. 585-592.
[29] M. Szummer and T. Jaakkola, "Information Regularization with Partially Labeled Data," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15. Vancouver, Canada, 2002, pp. 1025-1032.
[30] M. Flickher, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D.Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer*, vol. 28, pp. 23-32, 1995.
[31] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain., "Content-Based Image Retrieval at the End of Early Years," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1349-1380, 2000.

Dell Zhang is a research fellow in the National University of Singapore under the Singapore-MIT Alliance (SMA). He has received his BEng and PhD in Computer Science from the Southeast University, Nanjing, China. His primary research interests include machine learning, data mining, and information retrieval.

Wee Sun Lee is an Assistant Professor at the Department of Computer Science, National University of Singapore, and a Fellow of the Singapore-MIT Alliance (SMA). He obtained his Bachelor of Engineering degree in Computer Systems Engineering from the University of Queensland in 1992, and his PhD from the Department of Systems Engineering at the Australian National University in 1996. He is interested in computational learning theory, machine learning and information retrieval.