Semi-Supervised Learning with Very Few Labeled Training Examples

Zhi-Hua Zhou[1] **De-Chuan Zhan**[1] **Qiang Yang**[2]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Department of Computer Science & Engineering, Hong Kong University of Science & Technology, China
{zhouzh, zhandc}@lamda.nju.edu.cn qyang@cse.ust.hk

Abstract

In semi-supervised learning, a number of labeled examples are usually required for training an initial *weakly useful predictor* which is in turn used for exploiting the unlabeled examples. However, in many real-world applications there may exist very few labeled training examples, which makes the weakly useful predictor difficult to generate, and therefore these semi-supervised learning methods cannot be applied. This paper proposes a method working under a two-view setting. By taking advantages of the correlations between the views using canonical component analysis, the proposed method can perform semi-supervised learning with only one labeled training example. Experiments and an application to content-based image retrieval validate the effectiveness of the proposed method.

Introduction

In real-world applications of machine learning, it is often the case that abundant *unlabeled* training examples are available, but the labeled ones are fairly expensive to obtain since labeling examples requires much human effort. As a consequence, semi-supervised learning, which attempts to exploit the unlabeled examples in addition to labeled ones, has attracted much attention.

One of the many approaches to semi-supervised learning is to first train a *weakly useful predictor*, which is then used in exploiting the unlabeled examples. Here a weakly useful predictor h of a function f on a distribution $\mathcal{D}$ is a function satisfying $\Pr_{\mathcal{D}}[h(\boldsymbol{x}) = 1] \geq \epsilon$ and $\Pr_{\mathcal{D}}[f(\boldsymbol{x}) = 1 | h(\boldsymbol{x}) = 1] \geq \Pr_{\mathcal{D}}[f(\boldsymbol{x}) = 1] + \epsilon$ for some $\epsilon > 1/poly(n)$ where n is related to the description length of the examples (Blum & Mitchell 1998). In order to generate the initial weakly useful predictor, a number of labeled examples are needed.

However, in many real-world applications, the available labeled training examples may be very few. For example, in content-based image retrieval (CBIR), a user usually poses an example image as a query and asks a system to return similar images. In this case there are many unlabeled examples, i.e. images that exist in a database, but there is only one labeled example, i.e. the query image. Another example is online web-page recommendation. When a user is surfing the internet, he may occasionally encounter an interesting web page and may want the system bring him to similarly interesting web pages. It will be difficult to ask the user to identify more interesting pages as training examples because the user may not know where they are. In this example, although there are a lot of unlabeled examples, i.e. web pages on the internet, there is only one labeled example, i.e. the current interesting web page. In these cases, there is only one labeled training example to rely on. If the initial weakly useful predictor cannot be generated based on this single example, then the above-mentioned semi-supervised learning techniques cannot be applied.

Fortunately, the problem may be solvable; we show in this paper that, given two *sufficient views*, i.e. given that the data are described by two sets of attributes where each set is sufficient for learning, semi-supervised learning with only one labeled example is still feasible. This is because the correlation between these two views can provide some helpful information, which can be exploited by canonical component analysis. Experiments and an application to CBIR show that the proposed method, i.e. OLTV (learning with One Labeled example and Two Views), can work well in such situations.

We start the rest of this paper with a brief review on related works. Then, we propose OLTV and reports on our experiments, which is followed by the conclusion.

Related Works

Most of the current semi-supervised learning methods can be categorized into three main paradigms. In the first paradigm, a generative model such as a Naïve Bayes classifier or a mixture of Gaussians is used for the classifier, and the EM algorithm is employed to model the label estimation or parameter estimation process. Representative methods include (Miller & Uyar 1997; Nigam *et al.* 2000; Fujino, Ueda, & Saito 2005). In the second paradigm, the unlabeled data are used to regularize the learning process in various ways. For example, a graph can be defined on the data set where the nodes correspond to the examples while the edges encode the similarity between the examples, and then the label smoothness can be enforced over the graph as a regularization term. Representative methods include (Blum & Chawla 2001; Belkin & Niyogi 2004; Zhou, Schölkopf, & Hofmann 2005). Given the recent comprehensive reviews on semi-supervised learning (Chapelle, Schölkopf, & Zien 2006; Zhu 2006), in the following we focus our review on the third paradigm, i.e. co-training (Blum & Mitchell 1998).

Co-training works under a two-view setting, which trains two classifiers separately on two *sufficient and redundant views*. That is, two attribute sets are given, where each set is sufficient for learning and is conditionally independent of the other given the class label (Blum & Mitchell 1998). The predictions of each classifier on unlabeled examples are used to help augment the training set of the other classifier. Dasgupta et al. (2002) showed that the co-trained classifiers could make fewer generalization errors by maximizing their agreement over the unlabeled data. Later, Balcan et al. (2005) showed that given appropriately strong PAC-learners on each view, an assumption of *expansion* on the underlying data distribution is sufficient for co-training to succeed, which implies that the stronger assumption of *independence* between the two views is not necessary, and the existence of *sufficient views* is sufficient. Many variants of co-training have been developed, such as (Goldman & Zhou 2000; Zhou & Li 2005). In addition, co-training style algorithms have already been successfully applied to many applications (Sarkar 2001; Zhou, Chen, & Dai 2006).

As other semi-supervised learning methods, co-training style methods require a number of labeled training examples to be available. In particular, such methods cannot work well when there is only one labeled training example. There are many one-class methods that can be applied when there are only positive examples, but they require a set of labeled positive examples (Wang *et al.* 2005). In computer vision and pattern recognition areas, some methods have been developed to recognize an object class with one labeled example, but they still require a set of labeled examples of the other classes to be available (Fink 2005; Fleuret & Blanchard 2006). To the best of our knowledge, there is no semi-supervised learning method that can work with only one labeled training example.

The Proposed Method

Let $\mathcal{X}$ and $\mathcal{Y}$ denote two views, i.e. two attribute sets describing the data. Let $(\langle \boldsymbol{x}, \boldsymbol{y} \rangle, c)$ denote a labeled example where $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ are the two portions of the example, and c is the label. For simplifying our discussion, assume that $c \in \{0, 1\}$ where 0 and 1 denote negative and positive classes, respectively. Assume that there exist two functions $f_{\mathcal{X}}$ and $f_{\mathcal{Y}}$ over $\mathcal{X}$ and $\mathcal{Y}$, respectively, such that $f_{\mathcal{X}}(\boldsymbol{x}) = f_{\mathcal{Y}}(\boldsymbol{y}) = c$. Intuitively, this means that every example is associated with two views each contains sufficient information for determining the label of the example. Given $(\langle \boldsymbol{x}_0, \boldsymbol{y}_0 \rangle, 1)$ and a large number of unlabeled examples $\mathcal{U} = \{(\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle, c_i)\}$ ($i = 1, \cdots, l-1$; c_i is unknown), the task is to train a classifier to classify new examples.

Considering that the data are described by two sufficient views, some information concealed in the two views can be helpful if uncovered. Intuitively, some projections in these two views should have strong correlation with the ground truth. Actually, for either view, there should exist at least one projection which is correlated strongly with the ground truth, since otherwise this view can hardly be a *sufficient* one. Thus, if the correlated projections of these two views can be identified, they can help induce the labels of some unlabeled examples.

Canonical correlation analysis (CCA) (Hotelling 1936) is a statistical tool that can be used to identify the correlated projections between two views. CCA attempts to find two sets of basis vectors, one for each view, such that the correlation between the projections of these two views into the basis vectors are maximized. Formally, let $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{l-1})$ and $Y = (\boldsymbol{y}_0, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_{l-1})$. CCA finds projection vectors $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ such that the correlation coefficient between $\boldsymbol{w}_x^{\mathrm{T}} X$ and $\boldsymbol{w}_y^{\mathrm{T}} Y$ is maximized. That is,

$$\arg\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \left(\frac{\boldsymbol{w}_x^{\mathrm{T}} C_{xy} \boldsymbol{w}_y}{\sqrt{\boldsymbol{w}_x^{\mathrm{T}} C_{xx} \boldsymbol{w}_x \cdot \boldsymbol{w}_y^{\mathrm{T}} C_{yy} \boldsymbol{w}_y}} \right)$$

$$\text{w.r.t.} \begin{cases} \boldsymbol{w}_x^{\mathrm{T}} C_{xx} \boldsymbol{w}_x = 1 \\ \boldsymbol{w}_y^{\mathrm{T}} C_{yy} \boldsymbol{w}_y = 1, \end{cases}$$

where C_{xy} is the between-sets covariance matrix of X and Y, C_{xx} and C_{yy} are respectively the within-sets covariance matrices of X and Y. The corresponding Lagrangian is

$$L(\lambda_x, \lambda_y, \boldsymbol{w}_x, \boldsymbol{w}_y) = \boldsymbol{w}_x^{\mathrm{T}} C_{xy} \boldsymbol{w}_y$$
$$-\frac{\lambda_x}{2} \left(\boldsymbol{w}_x^{\mathrm{T}} C_{xx} \boldsymbol{w}_x - 1 \right) - \frac{\lambda_y}{2} \left(\boldsymbol{w}_y^{\mathrm{T}} C_{yy} \boldsymbol{w}_y - 1 \right).$$

By taking derivatives to $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, we get

$$\partial L / \partial \boldsymbol{w}_x = C_{xy} \boldsymbol{w}_y - \lambda_x C_{xx} \boldsymbol{w}_x = 0 \qquad (1)$$

$$\partial L / \partial \boldsymbol{w}_y = C_{yx} \boldsymbol{w}_x - \lambda_y C_{yy} \boldsymbol{w}_y = 0. \qquad (2)$$

By subtracting $\boldsymbol{w}_y \times$ Eq. 2 from $\boldsymbol{w}_x \times$ Eq. 1, we get

$$\begin{aligned} 0 &= \boldsymbol{w}_x^{\mathrm{T}} C_{xy} \boldsymbol{w}_y - \lambda_x \boldsymbol{w}_x^{\mathrm{T}} C_{xx} \boldsymbol{w}_x \\ &\quad - \boldsymbol{w}_y^{\mathrm{T}} C_{yx} \boldsymbol{w}_x + \lambda_y \boldsymbol{w}_y^{\mathrm{T}} C_{yy} \boldsymbol{w}_y \\ &= \lambda_y \boldsymbol{w}_y^{\mathrm{T}} C_{yy} \boldsymbol{w}_y - \lambda_x \boldsymbol{w}_x^{\mathrm{T}} C_{xx} \boldsymbol{w}_x \\ &= \lambda_y - \lambda_x. \end{aligned}$$

Let $\lambda = \lambda_x = \lambda_y$ and assume that C_{yy} is invertible, Eq. 3 can be derived from Eq. 2.

$$\boldsymbol{w}_y = \frac{1}{\lambda} C_{yy}^{-1} C_{yx} \boldsymbol{w}_x \qquad (3)$$

Substituting Eq. 3 into Eq. 1 gives the generalized eigenvalue problem in Eq. 4, from which $\boldsymbol{w}_x$ can be solved. Then, the corresponding $\boldsymbol{w}_y$ can be obtained from Eq. 3.

$$C_{xy} C_{yy}^{-1} C_{yx} \boldsymbol{w}_x = \lambda^2 C_{xx} \boldsymbol{w}_x \qquad (4)$$

The classical CCA can only find linear correlations. In order to identify nonlinearly correlated projections between the two views, kernel extensions of CCA can be used (Hardoon, Szedmak, & Shawe-Taylor 2004). KCCA maps $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ to $\phi_x(\boldsymbol{x}_i)$ and $\phi_y(\boldsymbol{y}_i)$ respectively ($i = 0, 1, \cdots, l-1$), and then treats $\phi_x(\boldsymbol{x}_i)$ and $\phi_y(\boldsymbol{y}_i)$ as instances in running the CCA routine. Formally, let $S_x = (\phi_x(\boldsymbol{x}_0), \phi_x(\boldsymbol{x}_1), \cdots, \phi_x(\boldsymbol{x}_{l-1}))$ and $S_y = (\phi_y(\boldsymbol{y}_0), \phi_y(\boldsymbol{y}_1), \cdots, \phi_y(\boldsymbol{y}_{l-1}))$. The projection vectors $\boldsymbol{w}_x^{\phi}$ and $\boldsymbol{w}_y^{\phi}$ in the higher-dimensional kernel space can be re-written as $\boldsymbol{w}_x^\phi = S_x^{\mathrm{T}}\alpha$ and $\boldsymbol{w}_y^\phi = S_y^{\mathrm{T}}\beta$ $(\alpha, \beta \in \mathcal{R}^l)$. Thus, the objective function becomes

$$\arg\max_{\alpha,\beta} \frac{\alpha^{\mathrm{T}} S_x S_x^{\mathrm{T}} S_y S_y^{\mathrm{T}} \beta}{\sqrt{\alpha^{\mathrm{T}} S_x S_x^{\mathrm{T}} S_x S_x^{\mathrm{T}} \alpha \cdot \beta^{\mathrm{T}} S_y S_y^{\mathrm{T}} S_y S_y^{\mathrm{T}} \beta}}.$$

Assume that $\mathcal{K}_x(\boldsymbol{a}, \boldsymbol{b}) = \phi_x(\boldsymbol{a})\phi_x^{\mathrm{T}}(\boldsymbol{b})$ and $\mathcal{K}_y(\boldsymbol{a}, \boldsymbol{b}) = \phi_y(\boldsymbol{a})\,\phi_y^{\mathrm{T}}(\boldsymbol{b})$ are respectively the kernel functions on the two views, and $K_x = S_x S_x^{\mathrm{T}}$ and $K_y = S_y S_y^{\mathrm{T}}$ are the two corresponding kernel matrices. α can be solved from

$$(K_x + \kappa I)^{-1} K_y (K_y + \kappa I)^{-1} K_x \alpha = \lambda^2 \alpha, \qquad (5)$$

where I is the identity matrix and κ is used for regularization. Then, β can be solved from Eq. 6.

$$\beta = \frac{1}{\lambda}(K_y + \kappa I)^{-1} K_x \alpha \qquad (6)$$

Thus, for any $\langle \boldsymbol{x}^*, \boldsymbol{y}^* \rangle$, its projection $\langle P(\boldsymbol{x}^*), P(\boldsymbol{y}^*) \rangle$ can be obtained according to $P(\boldsymbol{x}^*) = \phi_x(\boldsymbol{x}^*)\boldsymbol{w}_x^\phi = \phi_x(\boldsymbol{x}^*)S_x^{\mathrm{T}}\alpha = \mathcal{K}_x(\boldsymbol{x}^*, X)\alpha$ and $P(\boldsymbol{y}^*) = \phi_y(\boldsymbol{y}^*)\boldsymbol{w}_y^\phi$ $\phi_y(\boldsymbol{y}^*)S_y^{\mathrm{T}}\beta = \mathcal{K}_y(\boldsymbol{y}^*, Y)\beta$.

A number of α (and corresponding λ) can be solved from Eq. 5, while for each α a unique β can be solved from Eq. 6. This means that besides the most strongly correlated pair of projections, we can also identify the correlations between other pairs of projections and the strength of the correlations can be measured by the values of λ. If the two views are conditionally independent given the class label, the most strongly correlated pair of projections should be in accordance with the ground-truth. However, in real-world applications the conditional independence rarely holds, and therefore, information conveyed by the other pairs of correlated projections should not be omitted.

Let m denote the number of pairs of correlated projections that have been identified, an instance $\langle \boldsymbol{x}^*, \boldsymbol{y}^* \rangle$ can be projected into $\langle P_j(\boldsymbol{x}^*), P_j(\boldsymbol{y}^*) \rangle$ $(j = 1, 2, \cdots, m)$. Then, in the jth projection, the similarity between an original unlabeled instance $\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle$ $(i = 1, 2, \cdots, l-1)$ and the original labeled instance $\langle \boldsymbol{x}_0, \boldsymbol{y}_0 \rangle$ can be measured, for example, by Eq. 7 (the next section will show that other kinds of similarity measures can also be used),

$$\begin{aligned} sim_{i,j} &= \exp\left(-d^2\left(P_j\left(\boldsymbol{x}_i\right), P_j\left(\boldsymbol{x}_0\right)\right)\right) \\ &\quad + \exp\left(-d^2\left(P_j\left(\boldsymbol{y}_i\right), P_j\left(\boldsymbol{y}_0\right)\right)\right), \end{aligned} \qquad (7)$$

where $d(\boldsymbol{a}, \boldsymbol{b})$ is the Euclidean distance between $\boldsymbol{a}$ and $\boldsymbol{b}$. Considering that $\langle \boldsymbol{x}_0, \boldsymbol{y}_0 \rangle$ is a positive instance, the coefficient shown in Eq. 8 delivers the confidence of $\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle$ being a positive instance.

$$\rho_i = \sum\nolimits_{j=1}^{m} \lambda_j sim_{i,j} \qquad (8)$$

Thus, several unlabeled examples with the highest and lowest ρ values can be selected, respectively, as the extra positive and negative examples. The number of labeled training examples is thus increased. Note that here *positive confidence* is measured by the similarity to the original labeled positive example. While the most similar examples are likely to be positive, the most dissimilar examples may or may not be negative if the positive class contains several subconcepts. Therefore, the extra positive examples induced by OLTV are more reliable than the extra negative examples.

Table 1: The OLTV algorithm

```
Parameter:
  $\mathcal{K}$: the kernel function
  $\gamma_c$: the parameter controlling the number of extra labeled
      examples to be induced for the class $c$
Process:
  $\mathcal{L}_P \leftarrow \{seed\}$, $\mathcal{L}_N \leftarrow \emptyset$.
  Identify all pairs of correlated projections, obtaining $\alpha_i$, $\beta_i$,
  and $\lambda_i$ by solving Eqs. 5 and 6 on $\mathcal{U} \cup \mathcal{L}_P$.
  For $j = 0, 1, \cdots, l-1$ do
    Project $\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle$ into the $m$ pairs of correlated projections.
  For $j = 1, 2, \cdots, l-1$ do  Compute $\rho_j$ according to Eq. 8.
  $\mathcal{P} \leftarrow \arg\max_{\gamma_+, \langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle} (\rho_j)$, $\mathcal{N} \leftarrow \arg\min_{\gamma_-, \langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle} (\rho_j)$.
  % Rank the unlabeled instances according to their $\rho$ values;
  % use the top-$\gamma_+$ instances as extra positive instances, and
  % the bottom-$\gamma_-$ instances as extra negative instances.
  For all $\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle \in \mathcal{P}$ do  $\mathcal{L}_P \leftarrow \mathcal{L}_P \cup \{(\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle, 1)\}$.
  For all $\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle \in \mathcal{N}$ do  $\mathcal{L}_N \leftarrow \mathcal{L}_N \cup \{(\langle \boldsymbol{x}_j, \boldsymbol{y}_j \rangle, 0)\}$.
  $\mathcal{L} \leftarrow \mathcal{L}_P \cup \mathcal{L}_N, \mathcal{U} \leftarrow \mathcal{U} - (\mathcal{P} \cup \mathcal{N})$.
Output: $\mathcal{L}, \mathcal{U}$.
```

The OLTV algorithm is summarized in Table 1, where $seed$ denotes the original labeled example $(\langle \boldsymbol{x}_0, \boldsymbol{y}_0 \rangle, 1)$. After running OLTV, existing semi-supervised learning methods can be executed, since there are now addtional *labeled* training examples. Considering that OLTV works under a two-view setting, we employ co-training (Blum & Mitchell 1998) to accomplish the learning task since co-training also works under the two-view setting, but note that other kinds of semi-supervised learning methods can also be used.

Experiments

To evaluate our proposed algorithm, we perform experiments on four real-world data sets. The *course* data set (Blum & Mitchell 1998) has two views and contains 1,051 examples, each corresponding to a web page. The task is to predict whether a web page is a *course page* or not. For this task, there are 230 positive examples (roughly 22%). The *ads12*, *ads13* and *ads23* data sets are derived from the *ads* data set (Kushmerick 1999). Each example in these data sets corresponds to an image on the web, and the task is to predict whether an image is used for advertisement or not. 983 examples are used, among which 138 examples are positive (roughly 14%). The *ads* data set has more than two views. The *ads12* data set uses the url view and $origurl$ view, *ads13* uses url view and $destination\text{-}url$ view, while *ads23* uses $origurl$ view and $destination\text{-}url$ view.

On each data set, we perform a 10-fold cross validation (CV). In each fold, among the training examples, one positive example is randomly picked out to be used as the labeled training example, while the remaining training data are used as the unlabeled training examples. The entire 10-fold CV process is repeated ten times with random data partitions.

A single classifier trained on the only labeled training example, denoted by *L-one*, and a single classifier trained on all the training examples that are associated with ground-truth labels, denoted by *L-all*, are tested in our experiments as baselines. Moreover, co-training which does not use OLTV is compared. As stated earlier, since there is only one labeled training example, the standard co-training algorithm (Blum & Mitchell 1998) could not work. Therefore, two enhanced versions, *CT-n* and *CT-g*, are tested. CT-n and CT-g are different in how they deal with the shortage of labeled training examples. CT-n utilizes a k-nearest neighbor strategy to get additional labeled examples, whereas CT-g works in a "cheating" manner by assuming that it can be provided with the ground-truth labels of more examples. Although CT-g is not feasible in real-world applications, it is helpful for us to know the performance of OLTV in the experiments.

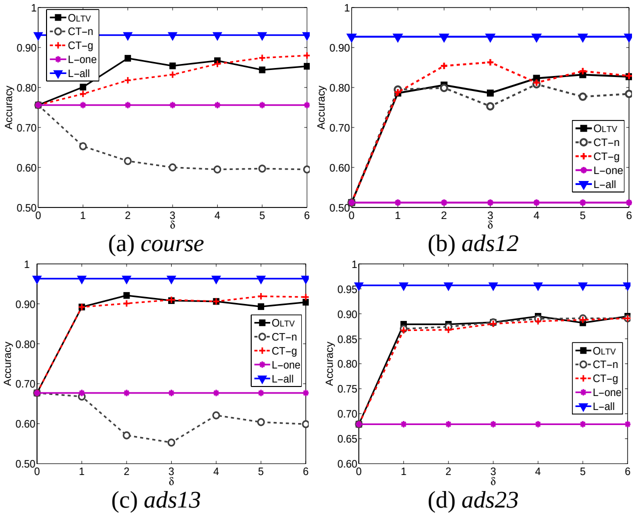
More specifically, besides the one labeled positive example, OLTV automatically induces $\gamma_+ = \delta \times p$ extra labeled positive examples and $\gamma_- = \delta \times q$ extra labeled negative examples [1]. Therefore, the co-training step at the end of OLTV has $1 + \delta \times (p + q)$ labeled examples. Besides the same labeled positive example, CT-n uses the $(\delta \times p)$-nearest examples of the labeled positive example as additional positive examples. Likewise, it uses $(\delta \times q)$-farthest examples as additional negative examples. Therefore, CT-n also uses $1 + \delta \times (p + q)$ labeled examples to launch the co-training process. Besides the same labeled positive example, CT-g is given $\delta \times p$ positive examples and $\delta \times q$ negative examples in random, where the ground-truth labels are provided (by "cheating"). Therefore, it also uses $1 + \delta \times (p + q)$ labeled examples to launch the co-training process.

The learning rounds of co-training used by OLTV, CT-g and CT-n are all set to 30 for *course* and 15 for *ads* data sets. Naïve Bayes classifier and J4.8 decision tree are used as alternatives to train the classifiers. The improvements on classification error obtained by exploiting unlabeled examples are presented in Table 2, which is computed by subtracting the final error from the initial error (equivalent to the error of L-one) and then divided by the initial error. Here δ is set to 6, and Gaussian kernel ($\sigma = 20$) is used by OLTV.

Table 2: Improvement (%) on classification error

Data set	Naïve Bayes			J48 Decision Tree		
	OLTV	CT-n	CT-g	OLTV	CT-n	CT-g
course	12.8	-21.3	16.4	8.5	-7.0	11.4
ads12	61.5	53.1	61.9	4.0	3.9	5.3
ads13	33.5	-11.5	35.4	27.1	10.4	26.7
ads23	31.8	31.2	31.2	38.9	37.1	40.4

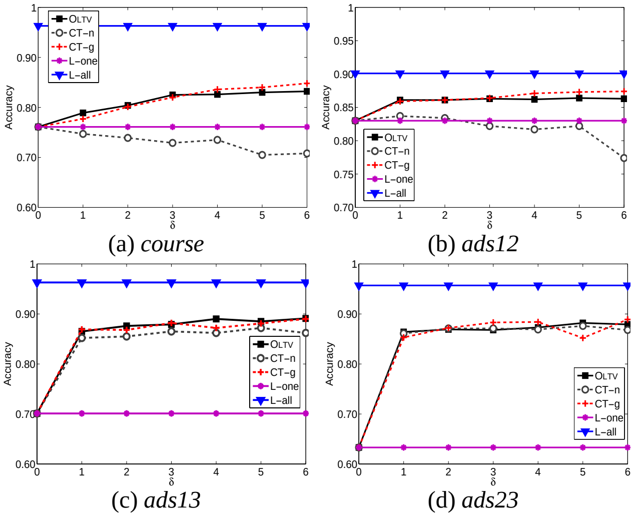
Table 2 shows that no matter which kind of classifier is used, OLTV can always improve the performance by exploiting unlabeled examples. CT-n often degenerates the performance, and even when it helps, the improvement is smaller than that of OLTV. It is impressive that the improvement of OLTV is comparable to that of CT-g which uses more ground-truth information.



(a) *course* (b) *ads12*

(c) *ads13* (d) *ads23*

Figure 1: Predictive accuracy with Naïve Bayes classifiers



(a) *course* (b) *ads12*

(c) *ads13* (d) *ads23*

Figure 2: Predictive accuracy with J48 decision trees

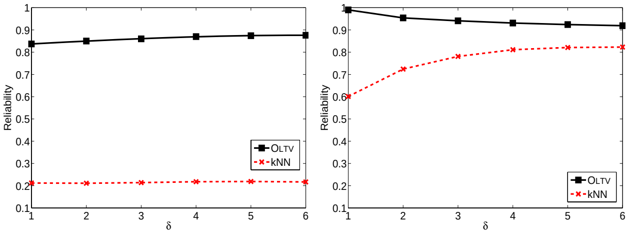
The predictive accuracy of OLTV (Gaussian kernel, $\sigma = 20$), CT-n, CT-g, L-one and L-all with different δ values are shown in Figures 1 and 2 [2]. Note that when $\delta = 0$, all the compared algorithms except L-all degenerate to L-one. The figures show that the predictive accuracy of OLTV is always better than that of L-one and usually better than that of CT-n. The difference between OLTV and CT-n on *ads12* and *ads23* are not so apparent as that on *ads13*, which suggests that the $oriurl$ view used by *ads12* and *ads23* but not by *ads13* is not

[1] Here δ is used to relate γ_c (the parameter of OLTV) to p and q (the parameters of co-training). Give p and q, the influence of γ_c on the performance of OLTV can be explored by studying the influence of δ. For convenience, assume that δ is a positive integer. According to (Blum & Mitchell 1998), in the experiments the ratio of $p : q$ is set to the positive/negative ratio of the data set, that is, $1 : 3$ is used on *course* while $1 : 6$ is used on the *ads* data sets.

[2] Actually we have evaluated another baseline, i.e. clustering the unlabeled data and then using the labeled example to label the clusters. Its performance is poor: the best accuracy is 0.6676, 0.2390, 0.2075 and 0.2232 on *course* and the *ads* data sets, respectively, which are much lower than the worst performance in the figures.

a sufficient view. Therefore, the correlation between the two views of *ads12* and *ads23* are not so helpful. It is impressive that the predictive accuracy of OLTV is comparable or sometimes even better than that of CT-g, which has been provided with ground-truth labels of $\delta \times (p + q)$ more examples [3].

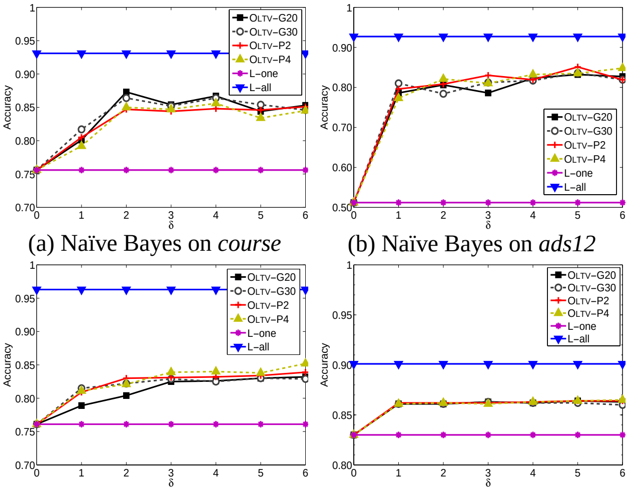
We conjecture that the success of OLTV can be attributed to the fact that it can induce additional labeled examples accurately by taking advantage of the two sufficient views. To evaluate this conjecture, we compare the *reliability* of OLTV (Gaussian kernel, $\sigma = 20$) to that of the kNN strategy used by CT-n. The reliability is defined as follows: assuming the labels for a unlabeled examples have been induced among which b of them are correct, the reliability is b/a. Due to the page limit, only the results on *course* and *ads12* are shown in Figure 3. It can be found that the reliability of the kNN strategy is far worse than that of OLTV. On all data sets, the reliability of OLTV is always higher than 80% and even often higher than 90%, which verifies our conjecture.



(a) *course* (b) *ads12*

Figure 3: The reliability of OLTV and kNN

To study the influence of the kernels on OLTV, we compare the performance of OLTV using Gaussian kernels ($\sigma = 20$ or 30) and Polynomial kernels ($power = 2$ or 4). Due to the page limit, only the results on *course* and *ads12* are shown in Figure 4. It can be found that OLTV is quite robust since employing different kernel functions or kernel parameters does not significantly change its good performance.



(a) Naïve Bayes on *course* (b) Naïve Bayes on *ads12*

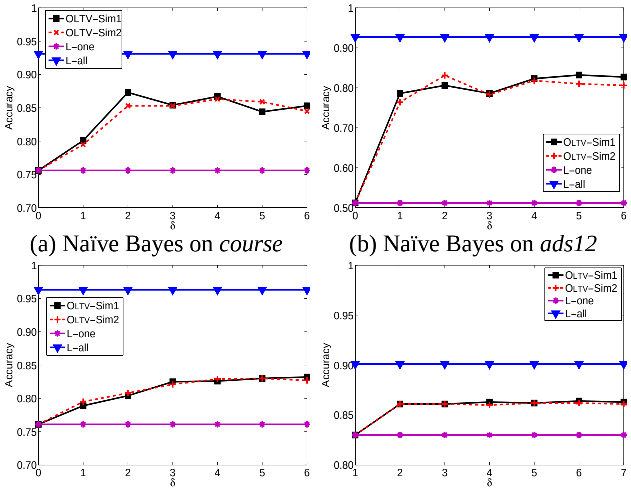
(c) J48 on *course* (d) J48 on *ads12*

Figure 4: OLTV using different kernels

In OLTV, a similarity measure (such as Eq. 7) is used to estimate the similarities between the unlabeled examples and the original labeled example under the correlated projections. As mentioned before, other kinds of similarity measures can also be used here. For example, Eq. 9 shows a measure where $\boldsymbol{a} \times \boldsymbol{b}$ denotes the inner product of $\boldsymbol{a}$ and $\boldsymbol{b}$, and $P_j(\boldsymbol{a})$ is the image of $\boldsymbol{a}$ in the j-th projection.

$$sim_{i,j} = P_j\left(\boldsymbol{x}_i\right) \times P_j\left(\boldsymbol{x}_0\right) + P_j\left(\boldsymbol{y}_i\right) \times P_j\left(\boldsymbol{y}_0\right) \qquad (9)$$

In order to study the influence of the similarity measures on OLTV, the performance of OLTV (Gaussian kernel, $\sigma = 20$) using Eqs. 7 and 9 are compared in Figure 5, where OLTV-sim1 and OLTV-sim2 denote OLTV using Eqs. 7 and 9, respectively. Due to the page limit, only the results on *course* and *ads12* are shown. It can be found that OLTV is quite robust since adopting Eqs. 7 or 9 does not significantly change its good performance.



(a) Naïve Bayes on *course* (b) Naïve Bayes on *ads12*

(c) J48 on *course* (d) J48 on *ads12*

Figure 5: OLTV using different similarity measures

Finally, we apply OLTV to CBIR. There are several previous studies that apply semi-supervised learning to CBIR (Tian *et al.* 2004; Zhou, Chen, & Dai 2006). However, all of them work in the *relevance feedback* process where a number of labeled examples are available. In contrast, we consider the initial query process where only one labeled example, i.e. the query itself, is available.

The experimental data are from (Zhang *et al.* 2005), where both the visual features and textual annotations are available. We perform experiments on 20 image classes each having 100 images (2,000 images totally), and regard the visual features and textual annotations respectively as two views. Here OLTV is employed to induce two positive images in addition to the query image, and as a result, three positive images can be used in querying the image database. Since no other semi-supervised learning methods can work in this situation, we compare OLTV with an ordinary CBIR process which does not exploit unlabeled images. Both methods use inner product to compute the similarity between image feature vectors. After repeating the tests on 10 different queries per image class (200 queries in total),

[3] CT-g uses 4 ($\delta = 1$) to 24 ($\delta = 6$) and 7 ($\delta = 1$) to 42 ($\delta = 6$) more labeled examples on *course* and *ads*, respectively.

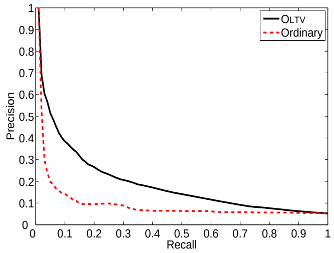


Figure 6: The average PR-graph

the average PR-graph is obtained, as shown in Figure 6. It can be found that the retrieval performance of standard retrieval process is apparently worse than that of OLTV. For example, when recall is 15%, the precision of OLTV (27.3%) is about 47% higher than that of the ordinary retrieval process (18.5%). This verifies that the OLTV method can be used to help improve the initial query performance of CBIR.

Conclusion

This paper shows that given two sufficient views, semi-supervised learning with a single labeled training example is feasible. By taking the advantage of the correlation between two views, the proposed OLTV method can effectively exploit the unlabeled examples for improving the learning performance. The success of OLTV implies that, if we can design two sufficient views for a concerned task, then asking the user to label only one example for the target class is sufficient for training a good predictor, which will make machine learning more readily available.

The OLTV method assumes that if two sufficient views are conditionally independent given the class label, the most strongly correlated pair of projections should be in accordance with the ground truth. Theoretical justification for this assumption is an important future work. Another important future issue is to explore whether semi-supervised learning with only one labeled example is feasible where there does not exist two sufficient views. Extending OLTV to multi-class cases is also worth studying. Moreover, it is interesting to combine OLTV with methods that can exploit unlabeled images during the relevance feedback process of CBIR, which can potentially boost the image-retrieval performance by exploiting images in the image databases.

Acknowledgements

Z.-H. Zhou and D.-C. Zhan were supported by NSFC (60635030, 60325207) and FANEDD (200343). Q. Yang was supported by HK-RGC (621606).

References

Balcan, M.-F.; Blum, A.; and Yang, K. 2005. Co-training and expansion: Towards bridging theory and practice. In *NIPS 17*. 89–96.

Belkin, M., and Niyogi, P. 2004. Semi-supervised learning on Riemannian manifolds. *Machine Learning* 56(1-3):209–239.

Blum, A., and Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 19–26.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, 92–100.

Chapelle, O.; Schölkopf, B.; and Zien, A., eds. 2006. *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

Dasgupta, S.; Littman, M.; and McAllester, D. 2002. PAC generalization bounds for co-training. In *NIPS 14*. 375–382.

Fink, M. 2005. Object classification from a single example utilizing class relevance metrics. In *NIPS 17*. 449–456.

Fleuret, F., and Blanchard, G. 2006. Pattern recognition from one example by chopping. In *NIPS 18*. 371–378.

Fujino, A.; Ueda, N.; and Saito, K. 2005. A hybrid generative/discriminative approach to semi-supervised classifier design. In *AAAI*, 764–769.

Goldman, S., and Zhou, Y. 2000. Enhancing supervised learning with unlabeled data. In *ICML*, 327–334.

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation* 16(12):2639–2664.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(4):321–377.

Kushmerick, N. 1999. Learning to remove internet advertisements. In *Agents*, 175–181.

Miller, D. J., and Uyar, H. S. 1997. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *NIPS 9*. 571–577.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2-3):103–134.

Sarkar, A. 2001. Applying co-training methods to statistical parsing. In *NAACL*, 95–102.

Tian, Q.; Yu, J.; Xue, Q.; and Sebe, N. 2004. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. In *ICME*, 1019–1022.

Wang, C.; Ding, C.; Meraz, R. F.; and Holbrook, S. R. 2005. PSoL: A positive sample only learning algorithm for finding non-coding RNA genes. *Bioinformatics* 22(21):2590–2596.

Zhang, R.; Zhang, Z.; Li, M.; Ma, W.-Y.; and Zhang, H. J. 2005. A probabilistic semantic model for image annotation and multi-modal image retrieval. In *ICCV*, 846–851.

Zhou, Z.-H., and Li, M. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans Knowledge and Data Engineering* 17(11):1529–1541.

Zhou, Z.-H.; Chen, K.-J.; and Dai, H.-B. 2006. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Trans Information Systems* 24(2):219–244.

Zhou, D.; Schölkopf, B.; and Hofmann, T. 2005. Semi-supervised learning on directed graphs. In *NIPS 17*. 1633–1640.

Zhu, X. 2006. Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI.