CROSS-DOMAIN LEARNING METHODS FOR HIGH-LEVEL VISUAL CONCEPT CLASSIFICATION

Wei Jiang, Eric Zavesky, Shih-Fu Chang

Department of Electrical Engineering
Columbia University
{wjiang,emz,sfchang}@ee.columbia.edu

Alex Loui

Kodak Research Labs
Eastman Kodak Company
{alexander.loui}@kodak.com

ABSTRACT

Exploding amounts of multimedia data increasingly require automatic indexing and classification, e.g. training classifiers to produce high-level features, or semantic concepts, chosen to represent image content, like car, person, etc. When changing the applied domain (i.e. from news domain to consumer home videos), the classifiers trained in one domain often perform poorly in the other domain due to changes in feature distributions. Additionally, classifiers trained on the new domain alone may suffer from too few positive training samples. Appropriately adapting data/models from an old domain to help classify data in a new domain is an important issue. In this work, we develop a new cross-domain SVM (CDSVM) algorithm for adapting previously learned support vectors from one domain to help classification in another domain. Better precision is obtained with almost no additional computational cost. Also, we give a comprehensive summary and comparative study of the state-of-the-art SVM-based cross-domain learning methods. Evaluation over the latest large-scale TRECVID benchmark data set shows that our CDSVM method can improve mean average precision over 36 concepts by 7.5%. For further performance gain, we also propose an intuitive selection criterion to determine which cross-domain learning method to use for each concept.

Index Terms— learning systems, feature extraction, adaptive systems, image processing

1. INTRODUCTION

There is a common issue for machine learning problems: the amount of available test data is large and growing, but the amount of labeled data is often fixed and quite small. Video data, labeled for semantic concept classification is no exception. For example, in high-level concept classification tasks (TRECVID [9]), new corpora may be added annually from unseen sources like foreign news channels or audio-visual archives. Ideally, one desires the same low error rates when reapplying models derived from previous source domain $\mathcal{D}^s$ to a new, unseen target domain $\mathcal{D}^t$, often referred to as domain adaptation or cross-domain learning. In this paper we tackle this challenging issue and make contributions in two folds. First, a new *Cross-Domain SVM* (*CDSVM*) algorithm is developed for adapting previously learned support vectors from source $\mathcal{D}^s$ to help detect concepts in target $\mathcal{D}^t$. Better precision can be obtained with almost no additional computational cost. Second, a comprehensive summary and comparative study of the state-of-the-art SVM-based cross-domain learning algorithms is given, and these algorithms are evaluated over the latest large-scale TRECVID benchmark data. Finally, a simple but effective criterion is proposed to determine if and which cross-domain method should be used.

The rest of this paper is organized as follows. Sec.2 gives an overview of many state-of-the-art SVM-based cross-domain learning methods, ordered in decreasing computational cost. Sec.3 introduces our CDSVM algorithm. Sec.4 and Sec.5 give the experimental results and the conclusion.

2. OVERVIEW

The cross-domain learning problem can be summarized as follows. Let $\mathcal{D}^t$ denote the *target data set*, which consists of two subsets: the labeled subset $\mathcal{D}^t_l$ and the unlabeled subset $\mathcal{D}^t_u$. Let $(\mathbf{x}_i, y_i)$ denote a data point where $\mathbf{x}_i$ is a d dimensional feature vector and y_i is the corresponding class label. In this work we only look at the binary classification problem, i.e., $y_i = \{+1, -1\}$. In addition to $\mathcal{D}^t$, we have a *source data set* $\mathcal{D}^s$ whose distribution is different from but related to that of $\mathcal{D}^t$. A binary classifier $f^s(\mathbf{x})$ has already been trained over this source data set $\mathcal{D}^s$. Our goal is to learn a classifier $f(\mathbf{x})$ to classify the unlabeled target subset $\mathcal{D}^t_u$.

Since $\mathcal{D}^t$ and $\mathcal{D}^s$ have different distributions, $f^s(\mathbf{x})$ will not perform well for classifying $\mathcal{D}^t_u$. Conversely, we can train a new classifier $f^t(\mathbf{x})$ based on $\mathcal{D}^t_l$ alone, but when the number of training samples $|\mathcal{D}^t_l|$ is small, $f^t(\mathbf{x})$ may not give robust performance. Since $\mathcal{D}^s$ is related to $\mathcal{D}^t$, utilizing information from source $\mathcal{D}^s$ to help classify target $\mathcal{D}^t_u$ should yield better performance. This is fundamental the motivation of cross-domain learning. In this section, we briefly summarize and discuss many state-of-the-art SVM-based cross-domain learning algorithms.

2.1. Standard SVM Applied in New Domain

Without cross-domain learning, the standard *Support Vector Machine* (*SVM*) [5] classifier can be learned based on the labeled subset $\mathcal{D}^t_l$ to classify the unlabeled set $\mathcal{D}^t_u$. Given a data vector $\mathbf{x}$, SVMs determine the corresponding label by the sign of a linear decision function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$. For learning non-linear classification boundaries, a kernel mapping ϕ is introduced to project data vector $\mathbf{x}$ into a high-dimensional feature space $\phi(\mathbf{x})$, and the corresponding class label is now given by the sign of $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$. The primary goal of an SVM is to find an optimal separating hyperplane that gives a low generalization error while separating the positive and negative training samples. This hyperplane is determined by giving the largest margin of separation between different classes, i.e. by solving the following problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{N^t_l} \epsilon_i \tag{1}$$
$$s.t.\ \ y_i\mathbf{w}^T\phi(\mathbf{x}_i)+b \geq 1-\epsilon_i,\ \epsilon_i \geq 0,\ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}^t_l$$

where ϵ_i is the penalizing variable added to each data vector $\mathbf{x}_i$; C determines how much error an SVM can tolerate.

One very simple way to perform cross-domain learning is to learn new models over all possible samples, called *Combined SVM* in this paper. The primary motivation for this method is that when the size of data in target domain is small, the target model will benefit from a high count of training samples present in $\mathcal{D}^s$ and should therefore be much more stable than a model trained on $\mathcal{D}^t$ alone. However, there is a large time cost for learning with this method due to the increased number of training samples from $|\mathcal{D}^t|$ to $|\mathcal{D}^s|+|\mathcal{D}^t|$.

2.2. Transductive Approach

To decrease generalization error in classifying unseen data $\mathcal{D}_u^t$ in the target domain, transductive SVM methods [4, 7] incorporate knowledge about the new test data into the SVM optimization process so that the learned SVM can accurately classify test data.

2.2.1. Transductive SVM (TSVM)

To find the optimal label estimation $\hat{y}_j$ over a test data vector $\hat{\mathbf{x}}_j$, $(\hat{\mathbf{x}}_j, \hat{y}_j) \in \mathcal{D}_u^t$, *Transductive SVM* (*TSVM*) [7] gives the optimal hyperplane by solving the following problem:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{N_l^t} \epsilon_i + \hat{C}\sum\nolimits_{j=1}^{N_u^t} \hat{\epsilon}_j \tag{2}$$

$$s.t.\ \ y_i\mathbf{w}^T\phi(\mathbf{x}_i)+b \geq 1-\epsilon_i,\ \epsilon_i \geq 0,\ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^t$$

$$\hat{y}_i\mathbf{w}^T\phi(\hat{\mathbf{x}}_j)+b \geq 1-\hat{\epsilon}_j,\ \hat{\epsilon}_j \geq 0,\ \forall(\hat{\mathbf{x}}_j, \hat{y}_j) \in \mathcal{D}_u^t$$

Eqn.(2) is not convex which it is generally hard to optimize. Many approximation methods have been used [3].

2.2.2. Localized SVM (LSVM)

Contrary to TSVMs that try to learn one general classifier by leveraging all the test data, the *Localized SVM* (*LSVM*) tries to learn one classifier for each test sample based on its local neighborhood. Given a test data vector $\hat{\mathbf{x}}_j$, we find its neighborhood in the labeled training set $\mathcal{D}_l^t$ based on similarity $\sigma(\hat{\mathbf{x}}_j, \mathbf{x}_i)$, $\mathbf{x}_i \in \mathcal{D}_l^t$: $\sigma(\hat{\mathbf{x}}_j, \mathbf{x}_i) = \exp\left(-\beta||\hat{\mathbf{x}}_j - \mathbf{x}_i||_2^2\right)$. β controls the size of the neighborhood, i.e. the larger the β, the less influence each distant data point has. An optimal local hyperplane is learned from test data neighborhoods by optimizing the following function:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{N_l^t} \sigma(\hat{\mathbf{x}}_j, \mathbf{x}_i)\epsilon_i \tag{3}$$

$$s.t.\ \ y_i\mathbf{w}^T\phi(\mathbf{x}_i)+b \geq 1-\epsilon_i,\ \epsilon_i \geq 0,\ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^t$$

As the result, the classification of a test sample only depends on the support vectors in its local neighborhood.

Transductive SVM approaches can be directly used for cross-domain learning by using $\mathcal{D}_l^t \cup \mathcal{D}^s$ to take the place of $\mathcal{D}_l^t$ in both Eqn.(2) and Eqn.(3). Their major drawback is the computational cost, especially for large-scale data sets. Let $\mathcal{O}^{ts}$ denote the time complexity of training a new classifier over the combined $\mathcal{D}_l^t \cup \mathcal{D}^s$. LSVM needs to train $|\mathcal{D}_u^t|$ classifiers, one for each test sample. Thus the complexity of LSVM is about $|\mathcal{D}_u^t|\mathcal{O}^{ts}$. In [3] the iterative training process for TSVM needs $P\mathcal{O}^{ts}$ complexity where P is the number of iterations. Approximation methods can be used to speed up the learning process by sacrificing accuracy [3, 4], but how to balance speed and accuracy is also an open issue.

2.3. Cross-domain Adaptation Approaches

In the cross-domain learning problem, the source data set $\mathcal{D}^s$ and the target data set $\mathcal{D}^t$ are highly related. The following cross-domain adaptation approaches investigate how to use source data to help classify target data.

2.3.1. Feature Replication

Feature replication combines all samples from both $\mathcal{D}^s$ and $\mathcal{D}^t$, and tries to learn *generalities* between the two data sets by replicating parts of the original feature vector, $\mathbf{x}_i$ for different domains. This method has been shown effective for text document classification over multiple domains [6]. Specifically, we first zero-pad the dimensionality of $\mathbf{x}_i$ from d to $d(N-1)$ where N is the total number of adaptation domains, and in our experiments $N=2$ (one source and one target). Next we transform all samples from all domains as:

$$\hat{\mathbf{x}}_i^s = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{0} \\ \mathbf{x}_i \end{bmatrix},\ \mathbf{x}_i \in \mathcal{D}^s \qquad \hat{\mathbf{x}}_i^t = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_i \\ \mathbf{0} \end{bmatrix}, \mathbf{x}_i \in \mathcal{D}^t$$

During learning, a model will be constructed that takes advantage of all possible training samples. Alike the combined method in section 2.1, this is most helpful when $\mathcal{D}^s$ can provide missing data for $\mathcal{D}^t$. However, unlike the combined method, learned SVs from the same domain as a test unlabeled sample (source-source or target-target) are given more preference by the the kernelized function of $\phi(\hat{\mathbf{x}}^s, \hat{\mathbf{x}}^s)$ or $\phi(\hat{\mathbf{x}}^t, \hat{\mathbf{x}}^t)$ compared to $\phi(\hat{\mathbf{x}}^t, \hat{\mathbf{x}}^s)$ because of the zero-padding operation. Unfortunately, due to the increase in dimensionality, there is also a large increase in model complexity and computation time during learning and evaluation of replication models.

2.3.2. Adaptive SVM

In [10], the *Adaptive SVM* (*A-SVM*) approach tries to adapt the a classifier $f^s(\mathbf{x})$, learned from $\mathcal{D}^s$ to classify the unseen target data set $\mathcal{D}_u^t$. In this approach, the final discriminant function is the average of $f^s(x)$ and the new "delta function" $\triangle f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$ learned from target set $\mathcal{D}_l^t$, i.e.,

$$f(\mathbf{x}) = f^s(\mathbf{x}) + \mathbf{w}^T\phi(\mathbf{x}) + b \tag{4}$$

where $\triangle f(\mathbf{x})$ aims at complementing $f^s(\mathbf{x})$ based on target $\mathcal{D}_l^t$. The basic idea of A-SVM is to learn a new decision boundary that is close to the original decision boundary (given by $f^s(\mathbf{x})$) as well as separating the target data. This new decision boundary can be obtained by solving problem:

$$\min_{\tilde{\mathbf{w}}} \frac{1}{2}||\tilde{\mathbf{w}}||_2^2 + C\sum\nolimits_{i=1}^{N_l^t} \epsilon_i,\ \tilde{\mathbf{w}} = [\mathbf{w}^T, b]^T \tag{5}$$

$$s.t.\ y_i(f^s(\mathbf{x}_i)+\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1-\epsilon_i,\ \epsilon_i \geq 0,\ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^t$$

The first term tries to minimize the deviation between the new decision boundary and the old one, and the second term controls the penalty of the classification error over the training data in the target domain.

One problem with this approach is the regularization constraint that the new decision boundary should not be deviated far from the source classifier, since equation (5) does not pursue large margin in learning with target data (note that $||\mathbf{w}||_2^2$ reflects the "delta function" but not margin in Eqn.(5)). It is a reasonable assumption when $\mathcal{D}^t$ is only incremental data for $\mathcal{D}^s$, i.e. $\mathcal{D}^t$ has similar distribution with $\mathcal{D}^s$. When $\mathcal{D}^t$ has a different distribution but comparable size than $\mathcal{D}^s$, such regularization constraint is problematic.

3. CROSS-DOMAIN SVM

In this work, we propose a new method called *Cross-Domain SVM* (*CDSVM*). Our goal is to learn a new decision boundary based on the target data set $\mathcal{D}_l^t$ which can separate the unknown data set $\mathcal{D}_u^t$, with the help of $\mathcal{D}^s$. Let $\mathcal{V}^s = \{(\mathbf{v}_1^s, y_1^s), \ldots, (\mathbf{v}_M^s, y_M^s)\}$ denote the support vectors which determine the decision boundary and $f^s(\mathbf{x})$ be the discriminant function already learned from the source domain. Learned support vectors carry all the information about $f^s(\mathbf{x})$; if we can correctly classify these support vectors, we can correctly classify

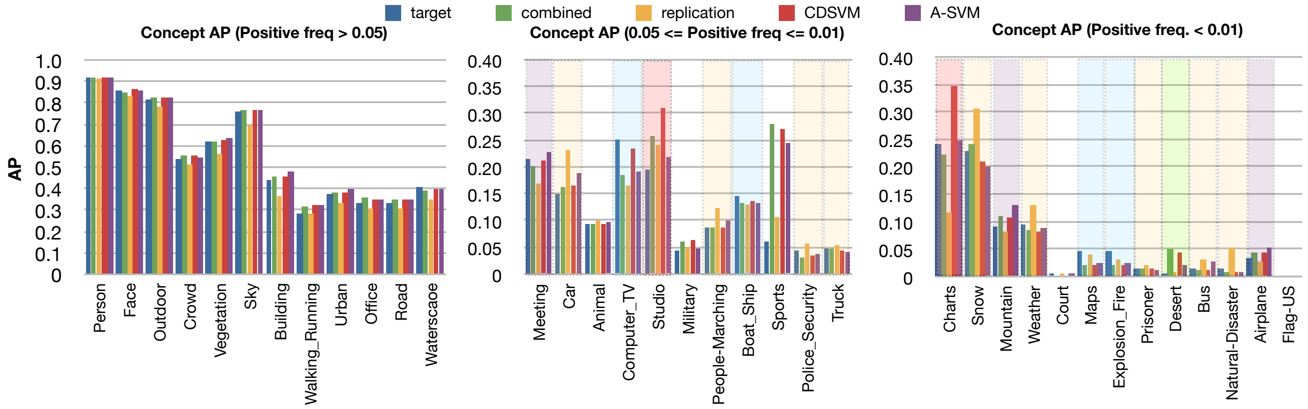

Fig. 1. Average precision vs. concept class for best methods; ordered by increasing frequency of positive $\mathcal{D}_l^t$ samples. Shaded concepts indicate the best method has a relative increase of at least 5% over all other methods.

the remaining samples from $\mathcal{D}^s$ except for some misclassified training samples. Thus our goal is simplified and analogous to learning an optimal decision boundary based on the target data set $\mathcal{D}_l^t$ which can separate the unknown data set $\mathcal{D}_u^t$ with the help of $\mathcal{V}^s$.

Similar to the idea of LSVM, the impact of source data $\mathcal{V}^s$ can be constrained by neighborhoods. The rationale behind this constraint is that if a support vector $\mathbf{v}_i^s$ falls in the neighborhood of target data $\mathcal{D}^t$, it tends to have a distribution similar to $\mathcal{D}^t$ and can be used to help classify $\mathcal{D}^t$. Thus the new learned decision boundary needs to take into consideration the classification of this support vector. Let $\sigma(\mathbf{v}_j^s, \mathcal{D}_l^t)$ denote the similarity measurement between source support vector $\mathbf{v}_j^s$ and the labeled target data set $\mathcal{D}_l^t$, our optimal decision boundary can be obtained by solving the following optimization problem:

$$\begin{aligned}&\min_{w} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{|\mathcal{D}_l^t|}\epsilon_i + C\sum\nolimits_{j=1}^{M}\sigma(\mathbf{v}_j^s, \mathcal{D}_l^t)\bar{\epsilon}_j \qquad (6)\\ &s.t.\;\; y_i(\mathbf{w}^T\phi(\mathbf{x}_i)-b)\geq 1-\epsilon_i,\; \epsilon_i\geq 0,\;\; \forall(\mathbf{x}_i, y_i)\in\mathcal{D}_l^t\\ &\qquad y_j^s(\mathbf{w}^T\phi(\mathbf{v}_j^s)-b)\geq 1-\bar{\epsilon}_j,\; \bar{\epsilon}_j\geq 0,\;\; \forall(\mathbf{v}_j^s, y_j^s)\in\mathcal{V}^s\end{aligned}$$

In CDSVM optimization, the old support vectors learned from $\mathcal{D}^s$ are adapted based on the new training data $\mathcal{D}_l^t$. The adapted support vectors are combined with the new training data to learn a new classifier. Specifically, let $\tilde{\mathcal{D}} = \mathcal{V}^s \cup \mathcal{D}_l^t$, Eqn.(6) can be rewritten as follows:

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{|\tilde{\mathcal{D}}|}\tilde{\sigma}(\mathbf{x}_i, \mathcal{D}_l^t)\epsilon_i \qquad (7)\\ &s.t.\;\; y_i(\mathbf{w}^T\phi(\mathbf{x}_i)-b)\geq 1-\epsilon_i,\; \epsilon_i\geq 0,\;\; \forall(\mathbf{x}_i, y_i)\in\tilde{\mathcal{D}}\\ &\tilde{\sigma}(\mathbf{x}_i,\mathcal{D}_l^t)=1,\; \forall(\mathbf{x}_i, y_i)\in\mathcal{D}_l^t,\; \tilde{\sigma}(\mathbf{x}_i,\mathcal{D}_l^t)=\sigma(\mathbf{x}_i, \mathcal{D}_l^t),\; \forall(\mathbf{x}_i, y_i)\in\mathcal{V}^s\end{aligned}$$

The dual problem of Eqn.(7) is as follows:

$$\begin{aligned}&\max_{\alpha_i} L_D = \sum\nolimits_{i=1}^{|\tilde{\mathcal{D}}|}\alpha_i - \frac{1}{2}\sum\nolimits_{i=1}^{|\tilde{\mathcal{D}}|}\sum\nolimits_{j=1}^{|\tilde{\mathcal{D}}|}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \qquad (8)\\ &s.t.\; \epsilon_i\geq 0,\; \mu_i\geq 0,\; 0\leq\alpha_i\leq C\tilde{\sigma}(\mathbf{x}_i, \mathcal{D}_l^t),\; y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b)\geq 1-\epsilon_i\\ &\alpha_i\left[y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b)-1+\epsilon_i\right]=0,\; \mu_i\epsilon_i=0,\;\; \forall(\mathbf{x}_i, y_i)\in\tilde{\mathcal{D}}\end{aligned}$$

Eqn.(8) is the same with the standard SVM optimization, with the only difference that:

$$\begin{aligned}&0\leq\alpha_i\leq C,\; \forall(\mathbf{x}_i, y_i)\in\mathcal{D}_l^t\\ &0\leq\alpha_i\leq C\sigma(\mathbf{x}_i, \mathcal{D}_l^t),\; \forall(\mathbf{x}_i, y_i)\in\mathcal{V}^s\end{aligned}$$

For support vectors from the source data set $\mathcal{D}^s$, weight σ penalizes those support vectors that are located far away from the new training samples in target data set $\mathcal{D}_l^t$.

Similar to A-SVM [10], we also want to preserve the discriminant property of the new decision boundary over the old source data $\mathcal{D}^s$, but our technique has a distinctive advantage: we do not enforce the regularization constraint that the new decision boundary is similar to the old one. Instead, based on the idea of localization, the discriminant property is only addressed over important source data samples that have similar distributions to the target data. Specifically, σ takes the form of a Gaussian function:

$$\sigma(\mathbf{v}_j^s, \mathcal{D}_l^t) = \frac{1}{|\mathcal{D}_l^t|}\sum\nolimits_{(\mathbf{x}_i, y_i)\in\mathcal{D}_l^t}\exp\left\{-\beta||\mathbf{v}_j^s - \mathbf{x}_i||_2^2\right\} \qquad (9)$$

β controls the degrading speed of the importance of support vectors from $\mathcal{V}^s$. The larger the β, the less influence of support vectors in $\mathcal{V}^s$ that are far away from $\mathcal{D}_l^t$. When β is very large, a new decision boundary will be learned solely based on new training data from $\mathcal{D}_l^t$. Also, when β is very small, the support vectors from $\mathcal{V}^s$ and the target data set $\mathcal{D}_l^t$ are treated equally and the algorithm is equivalent to training an SVM classifier over $\mathcal{D}_l^t \cup \mathcal{V}^s$ together. This is virtually equivalent to the combined SVM described in Sec.2.1. With such control, the proposed method is general and flexible, capturing conventional methods as special cases. The control parameter, β, can be optimized in practice via systematic validation experiments.

CDSVM has small time complexity. Let $\mathcal{O}^t$ denote the time complexity of training a new SVM based on labeled target $\mathcal{D}_l^t$. Since the number of support vectors from source domain, $|\mathcal{V}^s|$, is generally much smaller than the number of training samples in target domain, i.e., $|\mathcal{V}^s| \ll |\mathcal{D}_l^t|$, CDSVM trains an SVM classifier with $|\mathcal{V}^s|+|\mathcal{D}_l^t| \approx |\mathcal{D}_l^t|$ training samples, and this computational complexity is very close to $\mathcal{O}^t$.

4. EXPERIMENTS

In this work, we evaluated several algorithms over different parts of the TRECVID data set [1]. The source data set, $\mathcal{D}^s$, is a 41847 keyframe subset derived from the development set of TRECVID 2005, containing 61901 keyframes extracted from 108 hours of international broadcast news. The target data set, $\mathcal{D}^t$, is the TRECVID 2007 data set containing 21532 keyframes extracted from 60 hours of news magazine, science news, documentaries, and educational programming videos. We further partition the target set into training and evaluation partitions with 17520 and 4012 keyframes respectively. The TRECVID 2007 data set is quite different from TRECVID 2005 data set in program structure and production value, but they have similar semantic concepts of interest. All the keyframes are manually labeled for 36 semantic concepts, originally defined by LSCOM-lite [8], and in this work we train one-vs.-all classifiers.

Method	Train $\mathcal{D}^s$	Train $\mathcal{D}^t$	MAP
standard target	none	all	0.248
CDSVM	SVs	all	**0.263**
standard source	all	none	0.213
LSVM	regions	all	0.169
standard combined	all	all	0.257
replication	all	all	0.238

Table 1. Training data description and mean average precision (over 36 concepts) of each evaluated model ranked by increasing computation time, assuming $|\mathcal{D}^s| > |\mathcal{D}^t|$.

For each keyframe, 3 types of standard low-level visual features are extracted: grid-color moment (225 dim), Gabor texture (48 dim) and edge direction histogram (73 dim). These features are concatenated to form a 346-dim long feature vector to represent each keyframe. Such features, though relatively simple, have been shown effective in detecting scenes and large objects, and considered as part of standard features in high-level concept detection [1].

To guarantee model uniformity, we computed a single SVM RBF model for each concept and method with $C=1$ and $\gamma=\frac{1}{d}$ or 0.0029 for our experiments (d is the feature dimension). We used LIBSVM [2] for all computations with a modification to include sample independent weights, described in Eqn.(8).

4.1. Comparison of methods

Table 1 shows the mean average precision (MAP) of the classification task. Average precision is the precision evaluated at every relevant point in a ranked list averaged over all points; it is used here as a standard means of comparison for the TRECVID data set. Comparing MAP alone, the CDSVM method proposed in this work outperforms all other methods. This is significant not only because of the higher performance, but also because of lower computation complexity compared to the standard combined, replication, or LSVM methods. Improvements over the target model and the combined model are particularly encouraging and confirm our assumption that a judicious usage of data from the source domain is critical for robust target domain models. Not all the old samples are needed and inclusion of only source data support vectors is sufficient because each vector's influence is adequately customized.

4.2. Predicting method usage

While CDSVM has better average performance, further analysis demonstrates that it is not always the best choice for individual classes. Fig.1 gives the per-concept AP and is ordered such that frequency of positively labeled samples (as computed from $\mathcal{D}^t_l$) decreases from left to right. Intuitively, CDSVM will perform well when we have enough positive training samples in both $\mathcal{D}^s$ and $\mathcal{D}^t$. It is highly probable that support vectors from $\mathcal{D}^s$ are complementary to $\mathcal{D}^t$, which can be combined with $\mathcal{D}^t$ to get a good classifier. However, when training samples from both $\mathcal{D}^s$ and $\mathcal{D}^t$ are few, positive samples from both source and target will distribute sparsely in the feature space and it is more likely that the source support vectors are far from the target data. Thus, not much information can be obtained from $\mathcal{D}^s$ and we should not use cross-domain learning. Alternatively, with only a few positive target training samples and a very reliable source classifier $f^s(\mathbf{x})$, the source data may provide important missing data for the target domain. In such case, CDSVM will not perform well because target data is unreliable. In this second situation, the feature replication method, discussed in Sec.2.3.1 generally works well.

Based on the above analysis and empirical experimental results in Fig.1, a method predicting criterion is developed in Fig.2. With this criterion, we can increase our cross-domain learning MAP from 0.263 to 0.271, but one must cautiously interpret these MAP numbers. Though the overall MAP improvement is relatively small (about 3%), the improvements over the rare concepts is actually very significant. If we compute the MAP over only concepts with lower frequencies, the improvement is as large as 22%.

```
if $(freq(\mathcal{D}^t_+) > T^t_1) \cup (freq(\mathcal{D}^s_+) > T^s)$ then
    Selected model = CDSVM
else if $AP(\mathcal{D}^s) > MAP(\mathcal{D}^s)$ then
    Selected model = Feature Replication
else if $(freq(\mathcal{D}^t_+) < T^t_2) \cap (freq(\mathcal{D}^s_+) < T^s)$ then
    Selected model = SVM over Target Labeled Set $\mathcal{D}^t_l$
else
    Selected model = CDSVM
end if
```

Fig. 2. Method selection criterion. $freq(\mathcal{D}_+)$ is the frequency of positive samples in a data domain; $AP(\mathcal{D}^s)$ and and $MAP(\mathcal{D}^s)$ are the AP and MAP computed for source models on a validation set of source domain data; T^t_1, T^t_2 and T^s are thresholds empirically determined via experiments.

5. CONCLUSIONS

In this work we tackle the important cross-domain learning issue of adapting models trained and applied in different domains. We develop a novel and effective method for learning image classification models that work across domains even when the distributions are different and has training data is small. We also perform a systematic comparison of various cross-domain learning methods over a diverse and large video data set. To the best of our knowledge, this is the first work of such comprehensive evaluation. By analyzing the advantage and disadvantage of different cross-domain learning algorithms, a simple but effective criterion is proposed to determine when and which cross-domain learning methods should be used. Consistent performance improvement can be achieved in both overall MAP and individual AP over most concepts.

6. ACKNOWLEDGEMENTS

We would like to thank Jun Yang and his team for the availability of their software for the evaluation of the *ASVM* algorithm in our comparison experiments.

7. REFERENCES

[1] S.F. Chang, *et al.*, "Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction". In *NIST TRECVID workshop*, Gaithersburg, MD, 2005.

[2] C.C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines", *http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.

[3] Y. Chen, *et al.*, "Learning with progressive transductive support vector machines", *IEEE Intl. Conf. on Data Mining*, 2002.

[4] H.B. Cheng, *et al.*, "Localized support vector machine and its efficient algorithm", *Proc. SIAM Intl' Conf. Data Mining*, 2007.

[5] C. Cortes and V.Vapnik, "Support vector network", *Machine Learning*, vol.20, pp.273-297, 1995.

[6] H. Daumé III, "Frustratingly easy domain adaptation", *Proc. the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[7] A. Gammerman, *et al.*, "Learning by transduction", *Conf. Uncertainty in Artificial Intelligence*, pp.148-156, 1998.

[8] M. R. Naphade, *et al.*, "A Light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005," *IBM Research Technical Report*, 2005.

[9] A.F. Smeaton, *et al.*, "Evaluation campaigns and TRECVid", *Proc. ACM Intl' Workshop on MIR*, 2006.

[10] J. Yang, *et al.*, "Cross-domain video concept detection using adaptive svms", *ACM Multimedia*, 2007.