Chapter 1

Cross-Domain Learning for Semantic Concept Detection

Automatic semantic concept detection has become increasingly important to effectively index and search the exploding amount of multimedia content, such as those from the Web and TV broadcasts. The large and growing amount of unlabeled data in comparison with the small amount of labeled training data limits the applicability of classifiers based upon supervised learning. In addition, newly acquired data often have different distribution from the previous labeled data due to the changing characteristics of real-world events and user behaviors. For example, in concept detection tasks such as TRECVID [19], new collections may be added annually from unseen sources such as foreign news channels or audio-visual archives. There exists a non-negligible domain difference. To improve the semantic concept detection performance, these issues need to be addressed.

In this chapter, we investigate cross-domain learning methods that effectively incorporate information from available training resources in other domains to enhance concept detection in a target domain by considering the domain difference [1]. Our contribution lies in two folds. First, we develop three approaches to incorporate three types of information to assist the

[1]Data sets from different domains generally have different data distributions. It may be more appropriate to use the term "data distribution". We use "domain" to follow the notation in previous work.

target domain: the *Cross-Domain Support Vector Machine* (*CDSVM*) algorithm that uses previously learned support vectors; the *prediction-based method* that uses concept scores of the new data predicted by previously trained concept detectors; and the *Adaptive Semi-Supervised SVM* (*AS^{3}VM*) algorithm that incrementally updates previously learned SVM concept detectors to classify new target data. Second, we provide a comprehensive summary and comparative study of the state-of-the-art SVM-based cross-domain learning methods.

Cross-domain learning methods can be applied to classify semantic concepts in various types of data. For instance, we can assign semantic labels to images, videos, or *events* that are defined as groups of images and videos in this chapter. With regard to the three approaches we propose here, the CDSVM and AS^{3}VM algorithms are irrelevant to the specific data type. In other words, they directly work with feature vectors that are extracted to represent the underlying data. The prediction-based method, on the other hand, is developed for classifying event data, where the prediction hypotheses generated by previously trained concept detectors are used as features.

We extensively evaluate the proposed approaches over two scenarios by using four large-scale data sets: the TRECVID 2005 development data set containing 108 hours of videos in different languages from international broadcast news programs; the TRECVID 2007 data set containing 60 hours of videos from news magazines, science news, documentaries, and educational programming videos; Kodak's consumer benchmark set containing 1,358 videos from actual users representing different consumer groups; and Kodak's consumer event data set containing 1,972 events from actual users. In the first scenario, we use information from the TRECVID 2005 development data to enhance concept detection over the TRECVID 2007 data. Both data sets contain TV program videos, and we evaluate the cross-domain learning performance of using TV news videos to help classify TV documentary videos. In the second scenario, we use the TRECVID 2007 set to enhance concept detection over Kodak's consumer video and event data. We evaluate the prediction-based method and the AS^{3}VM algorithm, respectively. We aim to test the cross-domain learning performance when there is significant domain difference, *i.e.*, using TV programs to help classify consumer data. Experimental results show that compared with several state-of-the-art alternatives, the proposed approaches can significantly improve semantic classification in both scenarios.

1.1 Survey of Cross-Domain Learning for Concept Detection

We first define our learning problem. The goal is to classify a set of K concepts $C_1, \ldots, C_K$ in a data set $\mathcal{X}$ that is partitioned into a labeled subset $\mathcal{X}_L$ (with size $n_L \geq 0$) and an unlabeled subset $\mathcal{X}_U$ (with size $n_U > 0$), *i.e.*, $\mathcal{X} = \mathcal{X}_L \cup \mathcal{X}_U$. Each data point $\mathbf{x}_i \in \mathcal{X}_L$ is associated with a set of class labels y_{ik}, $k = 1, \ldots, K$, where $y_{ik} = 1$ or -1 indicates the presence or absence of concept C_k in $\mathbf{x}_i$. A data point $\mathbf{x}_i$ can be an image, a video, or an event (a set of images and videos grouped together). In addition to $\mathcal{X}$, we have a previous data set $\mathcal{X}^{old}$ (with size $n^{old} > 0$), whose data characteristics or distribution is different from but related to that of $\mathcal{X}$, *i.e.*, $\mathcal{X}$ and $\mathcal{X}^{old}$ are from different domains. A data point $\mathbf{x}_j \in \mathcal{X}^{old}$ can also be an image, a video, or an event. A set of classifiers (represented by a set of parameters Θ^{old}) have been learned using the old domain data $\mathcal{X}^{old}$ to detect another set of K^{old} concepts $C_1^{old}, \ldots, C_{K^{old}}^{old}$. Intuitively, there are several different scenarios to study. From the data type point of view, the target data $\mathcal{X}$ and the old-domain data $\mathcal{X}^{old}$ can have the same data type, or $\mathcal{X}$ and $\mathcal{X}^{old}$ can have different types of data. From the concept point of view, the target concepts $C_1, \ldots, C_K$ can be the same as the old-domain concepts $C_1^{old}, \ldots, C_{K^{old}}^{old}$, or $C_1, \ldots, C_K$ can be different from $C_1^{old}, \ldots, C_{K^{old}}^{old}$.

Cross-domain learning has been proposed recently as a technique to leverage information from the previous domain to enhance classification in the target domain. Such information can be selected data points or learned models from the previous domain. Several cross-domain learning methods have been developed for concept detection [5, 6, 11, 21], and they all deal with the scenario where $\mathcal{X}$ and $\mathcal{X}^{old}$ have the same type of data, and the target concepts $C_1, \ldots, C_K$ are the same as old-domain concepts $C_1^{old}, \ldots, C_{K^{old}}^{old}$ (*i.e.*, C_k is the same as C_k^{old}, and $K = K^{old}$). The CDSVM [10] and AS^{3}VM algorithms we develop in Section 1.2 and 1.3, respectively, deal with this scenario too. In Section 1.4, we describe our prediction-based method [9] that studies the scenario where data in $\mathcal{X}$ are events while data in $\mathcal{X}^{old}$ are images or videos, and $C_1, \ldots, C_K$ can be different from $C_1^{old}, \ldots, C_{K^{old}}^{old}$. In the following, we first briefly summarize the previous work, followed by some discussions of our approaches.

1.1.1 Standard SVM

Without cross-domain learning, the standard SVM classifier [20] can be learned based upon the labeled subset $\mathcal{X}_L$ to classify unlabeled data $\mathcal{X}_U$ and future unseen test samples. For

each concept C_k, given a datum $\mathbf{x}$ the SVM determines its corresponding label by the sign of a decision function $f(\mathbf{x})$. The optimal hyperplane gives the largest margin of separation between different classes and is obtained by solving the following problem:

$$\min_{\mathbf{f}\in\mathcal{H}} Q^{svm} = \min_{\mathbf{f}\in\mathcal{H}} \left\{ \gamma||\mathbf{f}||_2^2 + \frac{1}{n_L}\sum\nolimits_{i=1}^{n_L}(1-y_{ik}f(\mathbf{x}_i))_+ \right\}, \tag{1.1}$$

where $(1-y_{ik}f(\mathbf{x}_i))_+ = \max(0, 1-y_{ik}f(\mathbf{x}_i))$ is the hinge loss, $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{n_L})]^T$, $\mathbf{x}_i \in \mathcal{X}_L$, and γ controls the scale of the empirical error loss that the classifier can tolerate.

The simplest way to perform cross-domain learning is to learn new models over all possible training samples $\tilde{\mathcal{X}} = \mathcal{X}^{old} \cup \mathcal{X}_L$, *i.e.,* the *Combined SVM*. The primary motivation is that when the size of $\mathcal{X}_L$ is small, the target model will benefit from a high count of training samples present in $\mathcal{X}^{old}$ and, therefore, hopefully be more stable than a model trained on $\mathcal{X}_L$ alone. However, this method is computationally expensive if $\mathcal{X}^{old}$ is large. Also, the influence of new data in $\mathcal{X}_L$ may be overshadowed by the large amount of data in $\mathcal{X}^{old}$.

1.1.2 Semi-supervised Approaches

One intuitive way to improve the Combined SVM is to use semi-supervised learning. By incorporating knowledge about the unlabeled data $\mathcal{X}_U$ into the training process, semi-supervised learning methods [1, 4, 20, 23] can obtain better classifiers to classify test data. One most popular branch of semi-supervised learning is to use graph regularization [4, 23]. A weighted undirected graph $\mathcal{G}^d = (\mathcal{V}^d, E^d, \mathbf{W}^d)$ can be generated for the data set $\mathcal{X}$, where $\mathcal{V}^d$ is the vertices set and each node corresponds to a data point, E^d is the edges set, and $\mathbf{W}^d$ is weights set measuring the pairwise similarities among data points. To detect a concept C_k, a binary classifier is trained as follows. Under the assumption of label smoothness over $\mathcal{G}^d$, a discriminant function f is estimated to satisfy two conditions: the loss condition – it should be close to given labels y_{ik} for labeled nodes $\mathbf{x}_i \in \mathcal{X}_L$; and the regularization condition – it should be smooth on graph $\mathcal{G}^d$. Among graph-based methods, the *Laplacian SVM* (*LapSVM*) algorithm [1] is considered one of the state-of-the-art approaches in terms of both classification accuracy and the out-of-sample extension ability. Let $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{n_U+n_L})]^T$ be the vector of discriminant functions over $\mathcal{X}$. LapSVM solves the following problem:

$$\min_{\mathbf{f}\in\mathcal{H}} \{\gamma_A||\mathbf{f}||_2^2 + \gamma_I \mathrm{tr}\left(\mathbf{f}^T\mathbf{L}^d\mathbf{f}\right) + \frac{1}{n_L}\sum\nolimits_{\mathbf{x}_i\in\mathcal{X}_L}(1-y_{ik}f(\mathbf{x}_i))_+\}, \tag{1.2}$$

where $\mathbf{L}^d$ is the Laplacian matrix computed from $\mathbf{W}^d$.

Semi-supervised learning methods such as LapSVM can be applied directly to cross-domain learning problems by using $\tilde{\mathcal{X}} = \mathcal{X}^{old} \cup \mathcal{X}_L$ as the combined training data. However, due to the nonnegligible domain difference, the classifier may still be biased by $\mathcal{X}^{old}$. Also, such methods usually have high computation cost, especially for large-scale problems.

1.1.3 Feature Replication

The feature replication approach [5] uses all training samples from both $\mathcal{X}^{old}$ and $\mathcal{X}$, and tries to learn generalities between the two data sets by replicating parts of the original feature vector, $\mathbf{x}_i$, for different domains. 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 (in our experiments $N=2$). Next we transform all samples from all domains as:

$$\hat{\mathbf{x}}_i^{old} = \begin{bmatrix} \mathbf{x}_i^T & \mathbf{0} & \mathbf{x}_i^T \end{bmatrix}^T, \quad \mathbf{x}_i \in \mathcal{X}^{old} \; ; \qquad \hat{\mathbf{x}}_i^{new} = \begin{bmatrix} \mathbf{x}_i^T & \mathbf{x}_i^T & \mathbf{0} \end{bmatrix}^T, \quad \mathbf{x}_i \in \mathcal{X} \; .$$

During learning, for each concept C_k a model is constructed by using the transformed training data from both $\mathcal{X}^{old}$ and $\mathcal{X}$. However, due to the increase in feature dimensionality, there is a large increase in model complexity and computation time for both training and evaluation.

1.1.4 Domain Adaptive Semantic Diffusion (DASD)

The DASD algorithm [11] aims to improve classification of $\mathcal{X}_U$ by using affinity relationships among the K semantic concepts while considering the domain-shift problem. An undirected graph $\mathcal{G}^c = (\mathcal{V}^c, E^c, \mathbf{W}^{c,old})$ is defined to capture semantic concept affinity relations over the old domain. $\mathcal{V}^c$ is the vertices set, each node corresponding to a concept, E^c is the edges set, and $\mathbf{W}^{c,old}$ is the concept affinity matrix. Each entry W_{kl}^c gives the edge weight (representing the affinity relation) between C_k and C_l. Define the normalized graph Laplacian matrix $\mathbf{L}^{c,old}$:

$$\mathbf{L}^{c,old} = \mathbf{I} - \mathbf{D}^{c,old^{-1/2}} \mathbf{W}^{c,old} \mathbf{D}^{c,old^{-1/2}}, \tag{1.3}$$

where $\mathbf{D}^{c,old}$ is a diagonal matrix whose entries are row sums of $\mathbf{W}^{c,old}$. DASD makes an assumption of local smoothness over $\mathcal{G}^c$, *i.e.*, if two concepts have high similarity defined in $\mathcal{G}^c$, they frequently co-occur (or have similar discriminant functions) in data samples. Let $\mathbf{F} = [\mathbf{f}_1^c, \ldots, \mathbf{f}_K^c]$ be the discriminant functions over $\mathcal{X}$ for all concepts, $\mathbf{f}_k^c = [f_k(\mathbf{x}_1), \ldots, f_k(\mathbf{x}_{n_L+n_U})]^T$. The initial $\mathbf{F}$ is usually composed by discriminant functions generated by concept detectors that are trained from the old data $\mathcal{X}^{old}$. DASD solves the following problem to get refined $\tilde{\mathbf{F}}$ and $\mathbf{W}^{c,new}$ by iteratively updating the initial $\mathbf{F}$ and $\mathbf{W}^{c,old}$:

$$\min_{\tilde{\mathbf{F}}, \mathbf{W}^{c,new}} \operatorname{tr}(\tilde{\mathbf{F}}^T \mathbf{L}^{c,new} \tilde{\mathbf{F}})/2. \tag{1.4}$$

The major issue of DASD is the lack of the out-of-sample extension ability, *i.e.*, $\tilde{\mathbf{F}}$ is optimized over the available unlabeled data $\mathcal{X}_U$, and the learned results can not be easily applied to new unseen test data. Therefore, DASD does not have the incremental learning ability. This largely limits the applicability of DASD in many real problems.

1.1.5 Adaptive SVM (A-SVM)

The A-SVM algorithm [21] adapts classifiers Θ^{old} learned from the previous domain to classify $\mathcal{X}$ with the out-of-sample extension ability and incremental learning ability, without the requirement of retraining the entire model using data $\mathcal{X}^{old}$ from the previous domain. For a concept C_k, A-SVM adapts the old discriminant function f^{old} learned from $\mathcal{X}^{old}$ to classify the current data $\mathcal{X}$. The basic idea is to learn a new decision boundary that is close to the original decision boundary and can separate new labeled data. This is achieved by introducing a "delta function" $\Delta f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b$ to complement $f^{old}(\mathbf{x})$. The final discriminant function over a datum $\mathbf{x}$ is the average of $f^{old}(\mathbf{x})$ and $\Delta f(\mathbf{x})$. $\Delta f(\mathbf{x})$ can be obtained by minimizing the deviation between the new decision boundary and the old one, as well as minimizing the classification error over new labeled data.

One potential problem with this approach is the regularization constraint that the new decision boundary should not be deviated far from the old-domain classifier. It is a reasonable assumption when $\mathcal{X}$ only moderately deviates from $\mathcal{X}^{old}$, *i.e.*, $\mathcal{X}$ has similar distribution with $\mathcal{X}^{old}$. When $\mathcal{X}$ has a different distribution but comparable size than $\mathcal{X}^{old}$, such regularization can be problematic and can limit classification performance.

1.1.6 Overview of Our Methods

In this chapter, we develop three different cross-domain methods to use three different types of information from the old domain to help classification in the new target domain. For data incorporation, instead of using all training data from $\mathcal{X}^{old}$ like Combined SVM, we selectively use a fewer number of important data from the old domain to help classify new data. For classifier adaptation, instead of relying upon target labeled data $\mathcal{X}_L$ alone such as A-SVM, we incrementally update Θ^{old} by considering unlabeled data $\mathcal{X}_U$. For the prediction-based method, we incorporate prediction hypotheses generated by previously trained models from the old domain to enhance classification in the new domain.

1.2 CDSVM for Data Incorporation

In this section, we describe our CDSVM algorithm that learns a new decision boundary based upon the target labeled data $\mathcal{X}_L$ to separate the unlabeled data $\mathcal{X}_U$ and future unseen test data, with the help of $\mathcal{X}^{old}$. For a concept C_k, let $\mathcal{U}^{old} = \left\{(u_1^{old}, y_{1k}^{old}), \ldots, (u_{n^{s,old}}^{old}, y_{n^{s,old}k}^{old})\right\}$ denote the set of $n^{s,old}$ support vectors that determine the decision boundary and $f^{old}(\mathbf{x})$ be the discriminant function already learned from the old domain. The learned support vectors carry all of the information about $f^{old}(\mathbf{x})$; if we can correctly classify these support vectors, we can correctly classify the remaining samples from $\mathcal{X}^{old}$ except for some misclassified training samples. Therefore, instead of using all data from $\mathcal{X}^{old}$ directly, we only incorporate these support vectors $\mathcal{U}^{old}$ from the old domain. In addition, we make the assumption that the impact of each data in $\mathcal{U}^{old}$ can be constrained by neighborhoods. The rationale behind this constraint is that if a support vector u_j^{old} falls in the neighborhood of the target data $\mathcal{X}$, it tends to have a distribution similar to $\mathcal{X}$ and can be used to help classify $\mathcal{X}$. Thus the new learned decision boundary needs to take into consideration the classification of this support vector. Let $\sigma(u_j^{old}, \mathcal{X}_L)$ denote the similarity measurement between the old support vector u_j^{old} and the labeled target data set $\mathcal{X}_L$, our optimal decision boundary can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{n_L} \epsilon_i + C\sum\nolimits_{j=1}^{n^{s,old}} \sigma(u_j^{old}, \mathcal{X}_L)\bar{\epsilon}_j \qquad (1.5)\\ &s.t.\ y_{ik}(\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1-\epsilon_i, \epsilon_i \geq 0, \forall\ \mathbf{x}_i \in \mathcal{X}_L,\quad y_{jk}^{old}(\mathbf{w}^T\phi(u_j^{old})+b) \geq 1-\bar{\epsilon}_j, \bar{\epsilon}_j \geq 0, \forall\ u_j^{old} \in \mathcal{X}^{old}, \end{aligned}$$

where $\phi(\cdot)$ is a mapping function to map the original data into a high-dimension space.

In CDSVM optimization, the old support vectors learned from $\mathcal{X}^{old}$ are adapted based upon the new training data $\mathcal{X}_L$. The adapted support vectors are combined with the new training data to learn a new classifier. Let $\tilde{\mathcal{X}} = \mathcal{U}^{old} \cup \mathcal{X}_L$, Eqn. (1.5) can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{n_L+n^{s,old}} \tilde{\sigma}(\mathbf{x}_i, \mathcal{X}_L)\epsilon_i \qquad (1.6)\\ &s.t.\ y_{ik}(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1-\epsilon_i, \epsilon_i \geq 0, \forall\ \mathbf{x}_i \in \tilde{\mathcal{X}}\\ &\qquad \tilde{\sigma}(\mathbf{x}_i, \mathcal{X}_L) = 1, \forall\ \mathbf{x}_i \in \mathcal{X}_L, \tilde{\sigma}(\mathbf{x}_i, \mathcal{X}_L) = \sigma(\mathbf{x}_i, \mathcal{X}_L), \forall\ \mathbf{x}_i \in \mathcal{U}^{old}. \end{aligned}$$

The dual problem of Eqn. (1.6) is as follows:

$$\begin{aligned} &\max_{\alpha_i} L_D = \sum\nolimits_{i=1}^{n_L+n^{s,old}} \alpha_i - \frac{1}{2}\sum\nolimits_{i=1}^{n_L+n^{s,old}} \sum\nolimits_{j=1}^{n_L+n^{s,old}} \alpha_i\alpha_j y_{ik}y_{jk}K(\mathbf{x}_i, \mathbf{x}_j) \qquad (1.7)\\ &s.t.\ \epsilon_i \geq 0, \mu_i \geq 0, 0 \leq \alpha_i \leq C\tilde{\sigma}(\mathbf{x}_i, \mathcal{X}_L), \alpha_i\left[y_{ik}(\mathbf{w}^T\phi(\mathbf{x}_i)+b)-1+\epsilon_i\right] = 0, \mu_i\epsilon_i = 0, \forall\ \mathbf{x}_i \in \tilde{\mathcal{X}}, \end{aligned}$$

where $K(\cdot)$ is the kernel function and $K(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$. Equation (1.7) is the same as the standard SVM optimization, with the only difference that:

$$0 \leq \alpha_i \leq C, \forall\ \mathbf{x}_i \in \mathcal{X}_L, \qquad 0 \leq \alpha_i \leq C\sigma(\mathbf{x}_i, \mathcal{X}_L), \forall\ \mathbf{x}_i \in \mathcal{U}^{old}.$$

For support vectors from the old data set $\mathcal{X}^{old}$, weight σ penalizes those support vectors that are located far away from the new training samples in the target data set $\mathcal{X}_L$.

Similar to A-SVM [21], in CDSVM we also want to preserve the discriminant property of the new decision boundary over the old data $\mathcal{X}^{old}$, 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 upon the idea of localization, the discriminant property is addressed only over important old data samples that have similar distributions to the target data. Specifically, σ takes the form of a Gaussian function:

$$\sigma(u_j^{old}, \mathcal{X}_L) = \frac{1}{n_L} \sum\nolimits_{\mathbf{x}_i \in \mathcal{X}_L} \exp\{-\beta ||u_j^{old} - \mathbf{x}_i||_2^2\}. \tag{1.8}$$

Parameter β controls the degrading speed of the importance of support vectors from $\mathcal{U}^{old}$. The larger the β, the less influence of support vectors in $\mathcal{U}^{old}$ that are far away from $\mathcal{X}_L$. When β is very large, a new decision boundary will be learned solely based upon new training data from $\mathcal{X}_L$. When β is very small, the support vectors from $\mathcal{U}^{old}$ and the target data $\mathcal{X}_L$ are treated equally and the algorithm is equivalent to training an SVM over $\mathcal{U}^{old} \cup \mathcal{X}_L$ together. With such control, the proposed method is general and flexible. The control parameter, β, can be optimized in practice via systematic validation experiments. CDSVM has small time complexity. Let O_L denote the time complexity of training a new SVM based upon labeled target set $\mathcal{X}_L$. Because the number of support vectors from the old domain, $\mathcal{U}^{old}$, is generally much smaller than the number of training samples in target domain or the entire old data set, *i.e.*, $n^{s,old} \ll n_L$ and $n^{s,old} \ll n^{old}$, CDSVM trains an SVM classifier with $n^{s,old} + n_L \approx n_L$ training samples, and this computational complexity is very close to O_L. Therefore CDSVM is in general faster than Combined SVM or semi-supervised approaches.

1.3 AS^{3}VM for Incremental Classifier Adaptation

In this section, we study the scenario where there are only a few (or even none) training samples available in the new domain, *i.e.*, $\mathcal{X}_L$ is a small (or empty) set. In such a case, it is difficult to obtain a satisfactory classifier by using previous cross-domain learning methods. For example, both CDSVM and A-SVM rely mainly on $\mathcal{X}_L$ and will suffer from small sample

learning. Combined SVM or semi-supervised learning will be biased by $\mathcal{X}^{old}$ since the old training data dominate the entire training set. We develop an AS^{3}VM algorithm to accommodate this scenario. The main idea is to directly adapt the old classifiers Θ^{old} by using both $\mathcal{X}_L$ and $\mathcal{X}_U$, without retraining classifiers over all of the data. It is also desirable that such adaptation has the out-of-sample extension ability and can be conducted incrementally.

Before introducing the detailed AS^{3}VM algorithm, we first make our cross-domain learning problem more general. For each labeled data $\mathbf{x}_i^{new} \in \mathcal{X}_L$, we have a set of labels y_{ik}^{new}, $k=1,\ldots,K$. Instead of requiring $y_{ik}^{new}=1$ or -1, here y_{ik}^{new} can take three values, 1, 0, or -1, where $y_{ik}^{new}=1$ (-1) indicates that $\mathbf{x}_i^{new}$ is labeled as positive (negative) to the concept C_k, and $y_{ik}^{new}=0$ indicates that $\mathbf{x}_i^{new}$ is not labeled for C_k. That is, it is not necessary that each $\mathbf{x}_i^{new}$ is fully labeled to all K concepts. This is a frequent situation in reality because users commonly only annotate a few important concepts to a datum. Unless they are required to do so, users are reluctant to provide full annotation due to the burden of manual labeling.

1.3.1 Discriminative Cost Function

The previous concept detectors Θ^{old} are trained to separate data $\mathcal{X}^{old}$ in the old domain. To maintain this discriminative ability, we want the learned new models Θ^{new} to be similar to Θ^{old}. This is the same assumption used in some previous cross-domain methods such as A-SVM [21] described in Section 1.1.5. Therefore, the first part of the joint cost function that our AS^{3}VM minimizes is the following:

$$\min_{\Theta^{new}} ||\Theta^{new} - \Theta^{old}||_2^2 \,. \tag{1.9}$$

Specifically, SVMs are used as concept detectors from the old domain. According to the Representer Theorem [20], the discriminant function $f_k(\mathbf{x})$, which is learned from the old domain of a datum $\mathbf{x}$ for a concept C_k, is given as:

$$f_k(\mathbf{x}) = \sum\nolimits_{\mathbf{x}_i \in \mathcal{X}^{old}} \mu_{ik} K(\mathbf{x}_i, \mathbf{x}) = \mathbf{K}(\mathbf{x}; \mathcal{X}^{old})^T \mathbf{u}_k, \tag{1.10}$$

where $K(\cdot)$ is the kernel function, $\mathbf{K}(\mathbf{x}; \mathcal{X}^{old})$ is a vector composed by kernel functions of $\mathbf{x}$ against all data in $\mathcal{X}^{old}$, and $\mathbf{u}_k = [\mu_{1k}, \ldots, \mu_{n^{old}k}]^T$ (n^{old} is the size of $\mathcal{X}^{old}$). Define $\mathbf{U}^{old} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$. The $n^{old} \times K$ matrix $\mathbf{U}^{old}$ contains all parameters learned from the old domain to generate discriminant functions for classifying K concepts. Our goal is to learn a new matrix $\mathbf{U}^{new} = [\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_K]$ that is similar to $\mathbf{U}^{old}$. Thus Eqn. (1.9) can take the form:

$$\min_{\mathbf{U}^{new}} ||\mathbf{U}^{new} - \mathbf{U}^{old}||_2^2 \,, \tag{1.11}$$

where $||\cdot||_2$ is the Hilbert-Schmidt norm. The new discriminant function of classifying $\mathbf{x}$ for a concept C_k is given by:

$$\tilde{f}_k(\mathbf{x}) = K(\mathbf{x}; \mathcal{X}^{old})^T \tilde{\mathbf{u}}_k \ . \tag{1.12}$$

Now let us incorporate the new labeled data $\mathcal{X}_L$ into the above process. $\mathcal{X}_L$ can be added directly into the set of support vectors by assigning a set of parameters $\mathbf{u}_i^{new} = [\mu_{i1}^{new}, \ldots, \mu_{iK}^{new}]^T$ to each data sample $\mathbf{x}_i^{new} \in \mathcal{X}_L$, where:

$$\mu_{ik}^{new} = \begin{cases} \eta \cdot \min_i(\mu_{ik}), & y_{ik}^{new} = -1 \\ y_{ik}^{new} \cdot \max_i(\mu_{ik}), & \text{others} \end{cases} \ . \tag{1.13}$$

Parameter μ_{ik} is the parameter in original $\mathbf{U}^{old}$, and $0 \leq \eta \leq 1$ is a weight added to the negative new labeled samples. Due to the unbalancing between positive and negative samples in some real applications, *i.e.*, negative samples significantly outnumber positive ones for some concepts, we may need to treat positive and negative samples unequally. The weight μ_{ik}^{new} assigns more importance to the newly annotated data in $\mathcal{X}_L$ compared with old support vectors in $\mathbf{U}^{old}$. This is especially useful for small-size $\mathcal{X}_L$ since we need to emphasize the few newly labeled target data to obtain a good target classifier.

Let $\mathbf{U}^L = [\mathbf{u}_1^{new}, \ldots, \mathbf{u}_{n_L}^{new}]$. We can obtain the new amended parameter matrix $\hat{\mathbf{U}}^{old} = [\mathbf{U}^{old^T}, \mathbf{U}^{L^T}]^T$. Equation (1.11) can be directly rewritten to the following:

$$\min_{\mathbf{U}^{new}} ||\mathbf{U}^{new} - \hat{\mathbf{U}}^{old}||_2^2 \ , \tag{1.14}$$

which is the first part of the cost function AS^{3}VM optimizes.

1.3.2 Graph Regularization on Data Points

In order to use the large amount of unlabeled data in the new domain to assist classification, we incorporate the assumption of graph smoothness over data points from the semi-supervised learning, *i.e.*, close-by points in the feature space should have similar discriminant functions. Let undirected graph $\mathcal{G}^d = (\mathcal{V}^d, E^d, \mathbf{W}^d)$ denote the graph over $\mathcal{X}$ in the new domain, where $\mathcal{V}^d$ is the vertices set and each node corresponds to a data sample, E^d is the edges set, and $\mathbf{W}^d$ is the data affinity matrix. Each entry W_{ij}^d measures the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$. Then we have the following cost function:

$$\min_{\tilde{\mathbf{F}}} \frac{1}{2} \sum\nolimits_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} W_{ij}^d ||(\tilde{\mathbf{f}}_i^d / \sqrt{d_i^d}) - (\tilde{\mathbf{f}}_j^d / \sqrt{d_j^d})||_2^2. \tag{1.15}$$

$\tilde{\mathbf{F}} = [\tilde{\mathbf{f}}_1^d, \ldots, \tilde{\mathbf{f}}_{n_U+n_L}^d]^T$ contains the discriminant functions of $\mathcal{X}$ over all K concepts. $\tilde{\mathbf{f}}_i^d = [\tilde{f}_1(\mathbf{x}_i), \ldots, \tilde{f}_K(\mathbf{x}_i)]^T$ comprises discriminant functions over $\mathbf{x}_i$. d_i^d is the degree of graph $\mathcal{G}^d$

over node $\mathbf{x}_i$. By substituting Eqn. (1.12) into Eqn. (1.15), we obtain:

$$\min_{\mathbf{U}^{new}} \frac{1}{2} \mathrm{tr}\{\mathbf{U}^{newT}\mathbf{K}(\mathcal{X}^{old};\mathcal{X})\mathbf{L}^d\mathbf{K}(\mathcal{X};\mathcal{X}^{old})\mathbf{U}^{new}\}, \tag{1.16}$$

where $\mathbf{L}^d$ is the normalized graph Laplacian matrix:

$$\mathbf{L}^d = \mathbf{I} - \mathbf{D}^{d^{-1/2}}\mathbf{W}^d\mathbf{D}^{d^{-1/2}}. \tag{1.17}$$

$\mathbf{D}^d$ is a diagonal matrix whose entries are row sums of $\mathbf{W}^d$. $\mathbf{K}(\mathcal{X};\mathcal{X}^{old})$ is the kernel matrix of data set $\mathcal{X}$ against data set $\mathcal{X}^{old}$, and $\mathbf{K}(\mathcal{X};\mathcal{X}^{old}) = \mathbf{K}(\mathcal{X}^{old};\mathcal{X})^T$.

1.3.3 Solution

We can combine the cost functions Eqn. (1.14) and Eqn. (1.16) into a joint cost function to minimize by our AS^{3}VM algorithm:

$$\min_{\mathbf{U}^{new}} Q^{AS^3VM} =$$
$$\min_{\mathbf{U}^{new}} \left[||\mathbf{U}^{new} - \hat{\mathbf{U}}^{old}||_2^2 + (\lambda^d/2)\cdot\mathrm{tr}\{\mathbf{U}^{newT}\mathbf{K}(\mathcal{X}^{old};\mathcal{X})\mathbf{L}^d\mathbf{K}(\mathcal{X};\mathcal{X}^{old})\mathbf{U}^{new}\} \right]. \tag{1.18}$$

By optimizing Q^{AS^3VM} we can obtain a new parameter matrix $\mathbf{U}^{new}$ that constructs classifiers to classify all K concepts. By taking the derivative of the cost Q^{AS^3VM} with respect to $\mathbf{U}^{new}$ we can obtain:

$$\begin{aligned} \frac{\partial Q^{AS^3VM}}{\partial \mathbf{U}^{new}} = 0 \;\; &\Rightarrow \;\; 2\mathbf{U}^{new} - 2\hat{\mathbf{U}}^{old} + \lambda^d\mathbf{K}(\mathcal{X}^{old};\mathcal{X})\mathbf{L}^d\mathbf{K}(\mathcal{X};\mathcal{X}^{old})\mathbf{U}^{new} = 0 \\ &\Rightarrow \;\; \mathbf{U}^{new} = \left[I + \frac{\lambda^d}{2}\mathbf{K}(\mathcal{X}^{old};\mathcal{X})\mathbf{L}^d\mathbf{K}(\mathcal{X};\mathcal{X}^{old}) \right]^{-1} \hat{\mathbf{U}}^{old}. \end{aligned} \tag{1.19}$$

The AS^{3}VM algorithm has several advantages. First, AS^{3}VM can be conducted with or without the presence of new annotated data from the new domain. That is, when $n_L = 0$, $\hat{\mathbf{U}}^{old} = \mathbf{U}^{old}$, AS^{3}VM is still able to adapt old classifiers to the new domain by using Eqn. (1.19). This is in comparison to most previous domain-adaptive methods that rely upon new annotated data. Second, AS^{3}VM allows incremental adaptation. This extends the algorithm's flexibility in real applications because multimedia data sets (and their annotations) are usually accumulated incrementally. The major computation cost is from the matrix inversion, which is about $O((n^{old})^3)$.

1.4 Prediction-based Concept Score Incorporation

In this section, we develop a cross-domain learning system to adapt concept scores predicted by previously trained concept detectors from the old domain to the new domain. In the above two sections, the CDSVM and AS^{3}VM algorithms apply to the situation where both

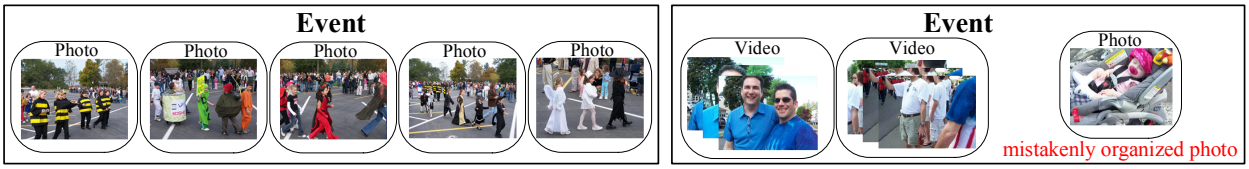

Figure 1.1: Two event data taken for different "parade" events, which have quite different visual appearances. These events are generated by an automatic albuming system, and in the event on the right a photo irrelevant to "parade" is mistakenly organized into this event.

the target data $\mathcal{X}$ and the old data $\mathcal{X}^{old}$ have the same data type, and the target concepts are the same as the old-domain concepts, *i.e.*, they work with feature vectors extracted from underlying data points, and such data points can be images, videos, or events. Different from these two methods, here we study the scenario where the target data $\mathcal{X}$ are events while the old data $\mathcal{X}^{old}$ are images or videos. An event is defined as a set of photos and/or videos that are taken within a common period of time, and have similar visual appearance. For example, an event can be composed by photos and videos taken by any user at the 2009 commencement of a university. Events are generated from unconstrained photo and video collections, by an automatic content management system, *e.g.*, an automatic albuming system. We want to assign one or multiple semantic labels to each event to describe its content, such as "wedding" and "graduation". In other words, the old domain data type is a building element of the target domain data type, *i.e.*, images and/or videos are building elements of events. Therefore, semantic concept detectors previously trained based on the old domain data can generate prediction hypotheses over the target data, and such hypotheses can be used as features to represent the target data. As a result, the target concepts $C_1, \ldots, C_K$ that we want to label to the event data can be different from the old-domain concepts $C_1^{old}, \ldots, C_{K^{old}}^{old}$ for which previous detectors are trained to generate prediction hypotheses.

Semantic classification of events has several characteristics. First, an event can contain both photos and videos, and we need to process photos and videos simultaneously. Second, the algorithm needs to accommodate errors resulting from automatic albuming systems. For example, in Fig. 1.1[2], a photo irrelevant to "parade" is mistakenly organized into a "parade" event. Third, events taken by different users, although from the same semantic category, can have quite diverse visual appearances, *e.g.*, as shown in Fig. 1.1, data from two "parade" events can look very different. In comparison, occasionally we do not have enough event data for robust learning, *e.g.*, in Kodak's consumer event collection we use in

[2]Fig. 1.1 is from [9] ©2009 Association for Computing Machinery, Inc. Reprinted by permission.

our experiment, there are only 11 "parade" events for training. The small sample learning difficulty may be encountered. This drives us to solicit help from cross-domain learning where we can borrow information from outside data sources to enhance classification.

1.4.1 Overview of Our System

Addressing the above characteristics, we develop a general two-step *Event-Level Feature* (*ELF*) learning framework, as described in Fig. 1.2[3]. In the first step each image (a photo or a video keyframe) is treated as a set of data points in an elementary-level feature space (*e.g.*, a concept score space at the image level or a low-level visual space at the region level). In the second step a unified ELF learning procedure is used to construct various ELFs based upon different elementary features. The ELF representation models each event as a feature vector, based upon which classifiers are directly built for semantic concept classification. The ELF representation is flexible to accommodate both photos and videos simultaneously, and is more robust to difficult or erroneous images from automatic albuming systems compared to the naive approach that uses image-level features to obtain classifiers straightforwardly.

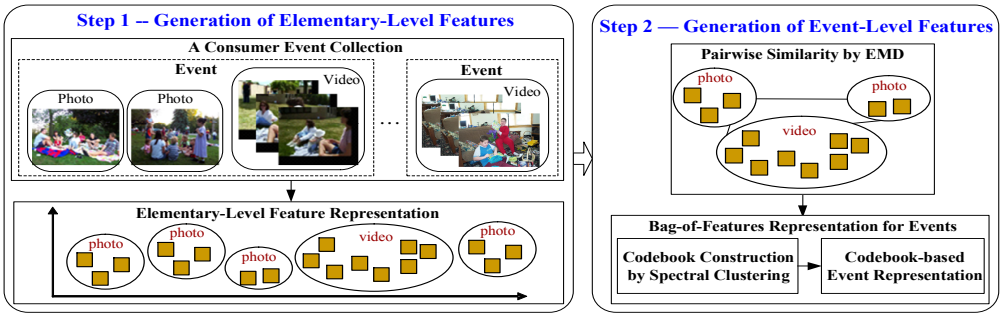

Figure 1.2: The general ELF learning framework. In the first step, each image (a photo or a video keyframe) is treated as a set of feature points in an elementary-level feature space, and then in the second step, an ELF representation can be constructed.

Using the general ELF learning framework, we conduct cross-domain and within-domain learning for semantic indexing in event data, as described in Fig. 1.3[4]. Complex target semantic concepts are usually generated by the concurrence of elementary constructive concepts. For example, "wedding" is a complex concept associated with people, park, *etc.*, evolving with a certain pattern. Based upon this idea, we adopt the PRED framework [5] for cross-domain learning. That is, a set of models detecting a set of elementary concepts $C_1^{old}, \ldots, C_{K^{old}}^{old}$ are built based upon the old data source, and are applied to the current data to generate concept occurrence predictions. Such predictions are then used as fea-

[3]Fig. 1.2 is from [9] ©2009 Association for Computing Machinery, Inc. Reprinted by permission.
[4]Fig. 1.3 is from [9] ©2009 Association for Computing Machinery, Inc. Reprinted by permission.

tures to represent the current data and to learn semantic concept detection models in the current domain. In practice, we incorporate two sets of concept detection scores from pre-trained models over two different old data sources, at both image and region level. They are: the TRECVID 2005 news video set [19] with a 374-concept LSCOM ontology [14]; and the LHI image-parsing ground-truth set with a 247-concept regional ontology [22]. Within-domain approaches use low-level visual features over entire images or image region segments as elementary-level features. The cross-domain and within-domain ELFs complement and cooperate with each other to improve classification.

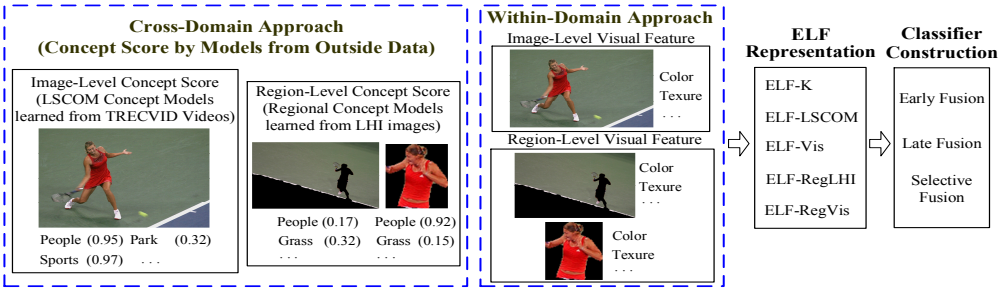

Figure 1.3: The overall framework of our concept detection approach over event data.

1.4.2 The ELF Learning Process

Assume that we have a collection of photos and videos from consumers, which is partitioned into a set of events. The partition is based upon the capture time of each photo/video and the color similarity between photos/videos, by using previously developed automatic albuming systems such as [13]. Let E^t be the t-th event, which contains m_p^t photos and m_v^t videos, and I_i^t and V_j^t be the i-th photo and j-th video in E^t, respectively. We define that both photos and videos are *data units*, represented by $\mathbf{x}$. For example, event E_t contains $m^t{=}m_p^t{+}\tilde{m}_v^t$ data units. Our goal is to assign E^t with semantic categories $C_1, \ldots, C_K$.

We first develop a *Bag-of-Features* (*BoF*) representation at the event level to describe each event as a feature vector, based upon which semantic concept detectors can be directly built. The BoF representation has been proven effective to detect generic concepts for images [18], where images are first represented by an orderless set of local descriptors (*e.g.*, SIFT features), and then through clustering a middle-level vocabulary is constructed. Visual words in the vocabulary are treated as robust and denoised terms to describe images.

In our event classification problem, for each semantic concept C_k, *e.g.*, "wedding", we have M events $E^1, \ldots, E^M$ that contain this concept. A vocabulary can be constructed by

clustering all data units from these M events into N words. Each word can be treated as a pattern that is a common characteristic for describing all events that contain C_k. To accommodate both photo and video data units, the similarity-based spectral clustering algorithm [16] is adopted to construct the vocabulary.

Specifically, the consumer video generally contains only one shot, and keyframes can be uniformly sampled from the videos. Let $I_{j,l}^t$ be the l-th keyframe in video V_j^t. Each photo I_i^t or keyframe $I_{j,l}^t$ can be represented as a set of feature points in the elementary-level feature space. For example, I_i^t is a single-point set with an image-level low-level visual feature $\mathbf{f}(I_i^t)$, or a multipoint set with region-level low-level visual features $\{\mathbf{f}(r_{i1}^t), \ldots, \mathbf{f}(r_{iG}^t)\}$ where each r_{ig}^t is a region from image I_i^t described by a feature vector $\mathbf{f}(r_{ig}^t)$.

By treating each data unit as a set of feature points in the elementary feature space, the *Earth Mover's Distance* (*EMD*) [17] can be adopted to measure the similarity between two data units (feature point sets). Note that there are many ways to compute the distance between two sets of feature points, *e.g.*, the maximum/minimum distance. These methods are easily influenced by noisy outliers, while EMD provides a more robust distance metric. EMD finds a minimum weighted distance among all pairwise distances between the two sets of feature points subject to weight-normalization constraints, and EMD allows partial matching between data units, which can alleviate the influence of noisy outliers. The pairwise EMD distance $D(\mathbf{x}_i, \mathbf{x}_j)$ between two data units $\mathbf{x}_i$, $\mathbf{x}_j$ can be converted to pairwise similarity based upon the Gaussian function: $S(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-D(\mathbf{x}_i, \mathbf{x}_j)/\beta\right)$, where β is the mean of all pairwise distances among training data units.

Given the pairwise similarity matrix over data units from the M events that contain semantic concept C_k, spectral clustering can be applied to find clusters of these data units. We adopt the algorithm developed in [16] where the number of clusters N can be determined automatically by analyzing eigenvalues of the similarity matrix. Each obtained data cluster is called a word, and all the clusters form a vocabulary. Let W_j be the j-th word, and let $S(\mathbf{x}, W_j)$ be the similarity of a datum $\mathbf{x}$ to word W_j calculated as the maximum similarity between $\mathbf{x}$ and the member data units in W_j. Assume that event E^t contains m^t data units in total, the entire event E^t can be represented by a BoF feature vector $\mathbf{f}_{bof}(E^t)$ as: $\mathbf{f}_{bof}(E^t) = [\max_{\mathbf{x}\in E^t} S(\mathbf{x}, W_1), \ldots, \max_{\mathbf{x}\in E^t} S(\mathbf{x}, W_N)]^T$.

1.4.3 Semantic Concept Classification with Multitype ELFs

The above ELF learning framework is very flexible. Different types of elementary-level features can be used to generate ELFs.

Cross-Domain ELFs

We further categorize the cross-domain ELFs as image-level or region-level, *i.e.*, concept detectors from external data sets are learned at the image or region level to generate the image-level or region-level elementary concept spaces.

Image-level concept space – We use the TRECVID 2005 development set [19] with a 374-concept LSCOM ontology [14] to generate a concept-score-based ELF at the image level. The LSCOM ontology contains 449 multimedia concepts related to objects, locations, people, and programs. The entire TRECVID 2005 development set is labeled to this ontology. By using visual features [3] over the entire image, *i.e.*, 5×5 grid-based color moments, Gabor texture, and edge direction histogram, 374 SVM concept detectors are learned based upon the TRECVID data, detecting 374 concepts with high-occurrence frequencies in LSCOM. These 374 concepts are the old-domain concepts $C_1^{old,trec}, \ldots, C_{374}^{old,trec}$, and we apply the 374 concept detectors to obtain the concept detection probabilities for each image I (a photo or a video keyframe) in the current event data set. These probabilities represent I in a concept space with a feature vector formed by concept scores $\mathbf{f}_c(I) = [p(C_1^{old,trec}|I), \ldots, p(C_{374}^{old,trec}|I)]^T$. Each photo is a single-point set and each video is a multipoint set in the concept space. Then the ELF learning process described in the second step of Fig. 1.2 can be used to generate the ELF over the LSCOM ontology, which is called *ELF-LSCOM*.

Region-level concept space – Region-level features provide detailed object information to describe the image content, which is complementary to global image-level features. In the regional approach, each image I is segmented into a set of regions $r_1, \ldots, r_G$, and each region can be represented by a feature vector in the elementary region-level feature space. Thus, both photos and videos are treated as multipoint sets, and the ELF learning procedure from the second step of Fig. 1.2 can be conducted to obtain ELF representations.

To generate region-level concept scores, we need external region-level concept detectors. In this work, the LHI image-parsing ground-truth data set (the free version) [22] is used to build region-level concept detectors. The data set contains images from 6 categories:

manmade object, natural object, object in scene, transportation, aerial image, and sport activity. These images are manually segmented and the regions are labeled to 247 concepts. Low-level visual features, *i.e.*, color moments, Gabor texture, and edge direction histogram, are extracted from each region. By using each region as one sample, SVM classifiers are trained to detect the 247 region-level concepts corresponding to the old-domain concepts $C_1^{old,LHI}, \ldots, C_{247}^{old,LHI}$. These detectors generate concept detection scores for each automatically segmented region in our event data. Then an ELF representation (*ELF-RegLHI*) can be learned based upon the region-level concept scores.

Within-Domain ELFs

The use of concept score space has been proven effective for semantic annotation by several previous works [7, 8]. However, low-level visual features are still indispensable, especially when we only have a limited concept ontology. Because in practice we cannot train a concept detector for every possible concept, low-level visual features can capture useful information not covered by the available concept detectors.

Within-domain visual-feature-based approaches can also be categorized as using image-level or region-level visual features. With image-level visual features, each image I is represented as a low-level visual feature vector. Then each photo is a single-point set and each video is a multipoint set, based upon which an ELF (*ELF-Vis*) can be generated. Specifically, we use the same low-level visual features as the ones to obtain image-level concept detection scores. Using region-level visual features, each region is represented as a low-level visual feature, and the entire image is a multipoint set in the regional feature space (as is a video), based upon which we generate an ELF (*ELF-RegVis*). In practice, we also use the same low-level visual features as the ones to obtain region-level concept detection scores. In addition, we use the concept detectors trained from Kodak's consumer benchmark video set with a 21-concept consumer ontology [12] to generate concept detection scores as elementary-level features to construct the ELF. We call this ELF representation *ELF-K*. This is treated as a within-domain learning approach, since both Kodak's benchmark videos and Kodak's event data are from the same consumer domain.

Classification with ELFs

By now we have five ELFs: ELF-K, ELF-LSCOM, ELF-RegLHI, ELF-Vis, and ELF-RegVis. Individual classifiers can be built over each ELF, and improved performance can be expected

if we appropriately fuse these ELFs. In early fusion, we concatenate these ELFs into a feature vector to train classifiers. In late fusion, we combine classifiers individually trained over ELFs. We can also use selective fusion, *i.e.*, forward feature selection. In selective early fusion, we gradually concatenate one more ELF at one time based upon the cross-validation error to choose the optimal combination of features. Similarly, in selective late fusion we gradually combine one more classifier trained over individual ELFs.

In Section 1.6.2, we will evaluate the concept detection performances of the prediction-based method over real event data from consumers, where both individual ELFs and their combinations are tested. From the result, the selective fusion can obtain more than 70% performance gain compared with individual ELFs.

1.5 Experiments: Cross-Domain Learning in TV Programs

We evaluate the CDSVM algorithm over two different TV program data sets. The first data set, $\mathcal{X}^{old}$, is a 41,847-keyframe set derived from the development set of TRECVID 2005, containing 61,901 keyframes extracted from 108 hours of international broadcast news videos. The target data set, $\mathcal{X}$, is the TRECVID 2007 data set containing 21,532 keyframes extracted from 60 hours of news magazine, science news, documentaries, and educational programming videos. We further partition the target set into training and test partitions with 17,520 and 4,012 keyframes, respectively. The partition is at the video level, *i.e.*, keyframes from the same video will be in the same set. The TRECVID 2007 data set is quite different from the TRECVID 2005 data set in program structure and production value, but they have similar semantic concepts of interests. All keyframes are manually labeled for 36 semantic concepts, originally defined by LSCOM-lite [15]. Both data sets are multi-label sets, *i.e.*, each keyframe may be labeled to multiple semantic concepts. One-vs.-all classifiers are trained to classify each concept. For each keyframe, three types of low-level visual features are extracted: grid color moments over 5×5 image grids, Gabor texture, and edge direction histogram. These features are concatenated to form a 346-dim feature vector to represent each keyframe. Such features, although relatively simple, have been shown effective in detecting generic concepts, and considered as part of standard features in semantic concept detection [3].

We compare CDSVM with several different alternatives in this section: the SVM trained using TRECVID 2005 data alone (SVM 05), the SVM trained using TRECVID 2007 data

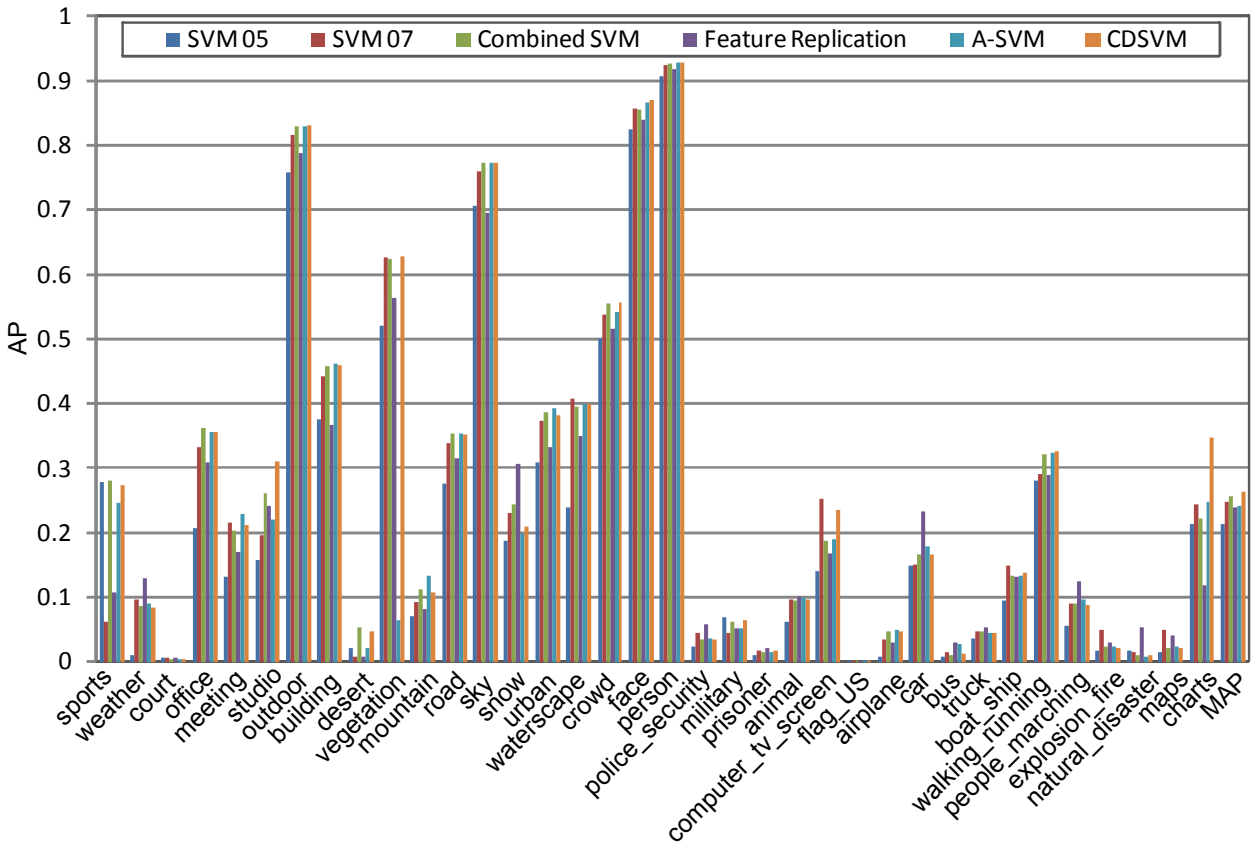

Figure 1.4: Result comparison: from TRECVID 2005 set to TRECVID 2007 set.

alone (SVM 07), the Combined SVM trained using the merged TRECVID 2005 and 2007 data, the Feature Replication method [5], and the A-SVM method [21]. To guarantee model uniformity, all SVM classifiers use the RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-\gamma||\mathbf{x}_i - \mathbf{x}_j||_2^2\}$ with $C = 1$ and $\gamma = 1/d$, where d is the feature dimension of $\mathbf{x}$. The LibSVM source code [2] is used and is modified to include sample independent weights, described in Eqn. (1.7).

Figure 1.4 shows the comparison of detection performances over 36 concepts by using different algorithms. The performance measurements are *Average Precision* (*AP*) and *Mean Average Precision* (*MAP*). AP is the precision evaluated at every relevant point in a ranked list averaged over all points; it is used here as a standard way of comparison for the TRECVID data set. MAP is the averaged AP across all concepts. From the figure, we can see that comparing MAP alone, the proposed CDSVM outperforms all other methods. This is significant not only because of the higher performance, but also because of the lower computation complexity. In addition, out of the three evaluated cross-domain learning methods, CDSVM is the only one that improves over the target model and the Combined SVM, while the

other two, both Feature Replication and A-SVM, can not. This phenomenon confirms our assumption that a judicious usage of data from the old domain is critical for learning robust target models. Because A-SVM pursues moderate modification of the old model instead of pursuing large-margin classification over the target domain, when the number of target training data is large to train a relatively good classifier, such moderate modification may be not as effective as retraining a target model directly. Feature Replication, on the other hand, uses all of the old data without selection, may be biased by the old data, and suffer from the high dimensionality of the replicated feature.

1.6 Experiments: From TV Programs to Consumer Videos

We conduct two sets of experiments to evaluate the AS^{3}VM algorithm and the prediction-based method over three data sets: the TRECVID 2007 data set, Kodak's consumer benchmark video set [12], and Kodak's consumer event data set [9]. Kodak's consumer benchmark set contains 1,358 videos from about 100 actual users representing different consumer groups. A total of 5,166 keyframes are sampled from these videos and are labeled to 21 consumer concepts. Kodak's event set contains 1,972 consumer events, which are generated from the automatic albuming system described in [13], and are labeled to 10 semantic categories. The details, such as definitions of these semantic categories and descriptions of the event data, can be found in [8].

1.6.1 AS^{3}VM for Semantic Concept Detection

Kodak's set and the TRECVID 2007 set are from different domains. Among the 36 concepts annotated over the TRECVID data, 5 concepts are similar to the consumer concepts annotated over Kodak's benchmark data. They are animal (animal), boat-ship (boat), crowd (crowd), people-marching (parade), and sports (sports), where concepts in parentheses are defined for Kodak's set. We adaptively apply the 5 SVM concept detectors trained over TRECVID 2007 data to Kodak's benchmark data by using the AS^{3}VM algorithm. The performance measures are AP and MAP. We evaluate two scenarios where we do not have new labeled data or have some labeled data, from Kodak's consumer set. Algorithms in these scenarios are marked by "(n)", and "(l)", respectively, *e.g.*, "(n) AS^{3}VM" and "(l) AS^{3}VM". Figure 1.5 shows the performance comparison in the first scenario where we

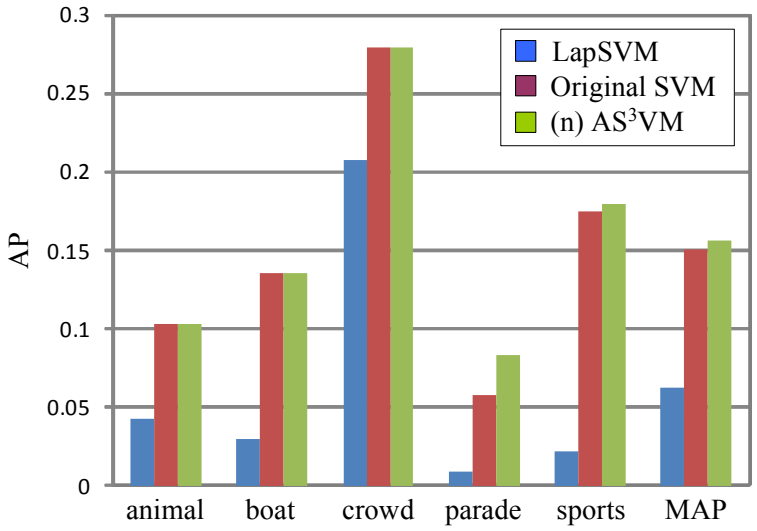

Figure 1.5: From TRECVID 2007 set to Kodak's benchmark set: without new annotation.

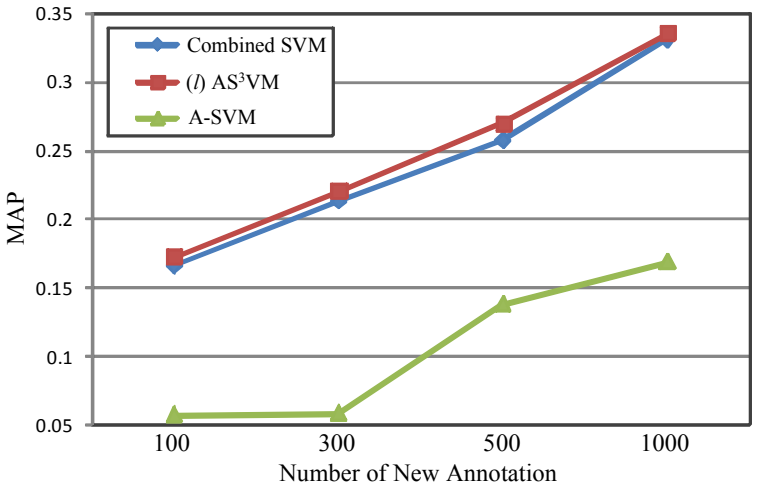

Figure 1.6: From TRECVID 2007 set to Kodak's benchmark set: with new annotations.

compare AS^{3}VM with semi-supervised LapSVM [1] and original SVM (directly applying TRECVID-based SVMs). For LapSVM, we treat the TRECVID 2007 data as training data and Kodak's consumer data as unlabeled data. This is one intuitive alternative of learning classifiers that use information from both data sets without new annotations. The results show that A^{3}SVM can improve the performance of original TRECVID-based SVMs by about 4% in terms of MAP on a relative basis. LapSVM, which treats both data sets as from the same distribution, does not perform well due to the non-negligible domain difference.

Figure 1.6 shows the performance comparison in the second scenario with different numbers of annotated data from the new domain. A set of randomly selected data in Kodak's

benchmark set are provided to users, and for each data one concept is randomly chosen for users to annotate. The annotation rate is pretty low, *i.e.*, from 0.4% to 4% when we have 100 to 1000 annotations compared with 5166×5 annotations to fully annotate the entire target set. Results in Figure 1.6 are the averaged results over 10 random runs. Here we compare AS^{3}VM with two other alternatives: the Combined SVM using all labeled data from both the TRECVID 2007 set and Kodak's benchmark set, and the cross-domain A-SVM [21] of adapting TRECVID-based SVMs to Kodak's data using new labeled data. The figure shows that AS^{3}VM can effectively improve the classification performance by outperforming the Combined SVM. In comparison, A-SVM can not improve detection because it updates classifiers only based upon the few labeled samples that are often biased. The results indicate the superiority of our method by both using information from unlabeled data and adapting classifiers to accommodate the domain change.

1.6.2 Prediction-based Method for Concept Detection in Events

From Kodak's consumer event set, a total of 1,261 events are randomly selected for training, the rest for testing. AP and MAP are still used as performance measures.

Figure 1.7 gives the individual AP and the overall MAP using different individual ELFs. From the result, different types of ELFs have different advantages in classifying different semantic categories. In general, image-level concept scores (ELF-K and ELF-LSCOM) perform well over complex semantic concepts such as "birthday", "parade", "picnic", "school activity", and "wedding", which are composed of many constructive concepts, *e.g.*, wedding consists of wedding gowns, suits, park, flowers, *etc.* The concept scores capture the semantic information about occurrences of these constructive concepts. On the other hand, ELF-Vis performs extremely well over semantic categories that are determined by only one or a few concepts, such as "animal", where the detection scores for other constructive concepts are not so helpful. Similarly, ELF-RegLHI performs well over complex semantic categories in general, and it works very well over those semantic categories having strong regional cues, *e.g.*, "individual sport" or "show", where detection of sport fields or stages helps greatly.

In terms of image-level concept scores, the large ontology (ELF-LSCOM) outperforms the small one (ELF-K), although concept detectors for the latter are trained with consumer videos that are more similar to our consumer event data than the TRECVID data. This

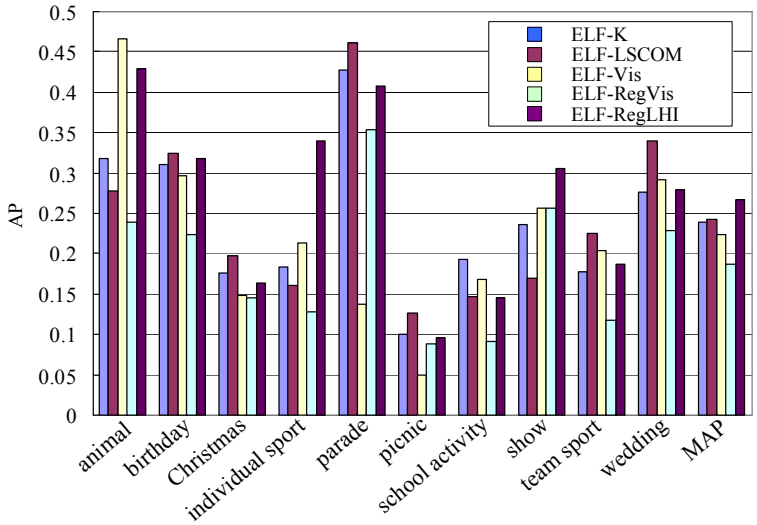

Figure 1.7: Performances of individual ELFs.

confirms that a large ontology can provide rich descriptors to represent the media content and a large external data source can be quite helpful. Specifically, ELF-LSCOM gets very good results over "parade", "team sport", and "wedding". This is because the TRECVID news videos and the LSCOM ontology provide good detectors for many constructive concepts related to parade (*e.g.*, protest, demonstration, *etc.*), sports (*e.g.*, basketball, football, *etc.*), and well-suited people (*e.g.*, corporate leader, government leader, and so on).

Figure 1.8 shows performances of different fusion methods, and the best individual ELF is also given for comparison. From the result, consistent performance improvements can be achieved over every semantic concept when we combine different ELFs by either early or late fusion, *i.e.*, about 35% gain in MAP compared to the best individual ELF. In addition, by selectively combining different ELFs, further performance gain can be obtained. Compared to the best individual ELF, the selective fusion can attain more than 70% MAP improvement.

1.7 Conclusion

We study the cross-domain learning issue of incorporating information from available training resources in other domains to enhance concept detection in a target domain. We develop three approaches: the CDSVM algorithm that uses previously learned support vectors; the AS^{3}VM algorithm that incrementally updates previously learned concept detectors; and the prediction-based method that uses concept scores predicted by previously trained concept detectors. Experiments over both TRECVID data from TV programs and Kodak's consumer

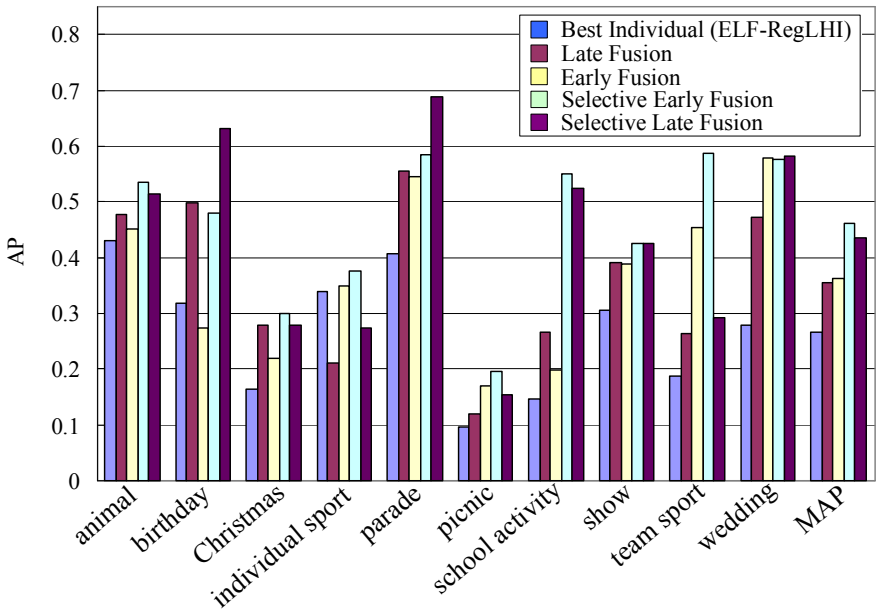

Figure 1.8: Performances of different fusion methods. Significant improvements can be achieved by selectively combining different ELFs.

videos demonstrate the effectiveness of our approaches.

In general, all three methods are developed to deal with relatively large domain differences. However, if the domain difference is very large, the prediction-based method is more robust than the other two. The reason is that compared to incorporating old data or updating old models, using concept scores of new data as features to train new classifiers is less sensitive to the distribution change between old and new domains. On the other hand, if there are very few training data available in the new domain, and the domain difference is not very dramatic, the AS^{3}VM algorithm tends to work better than the other two. This is because AS^{3}VM relies on both the old models and the new data structure to obtain updated classifiers while the other two mainly rely upon the insufficient new training data.

In addition, both CDSVM and AS^{3}VM deal with the scenario where we have the same type of data in both the target and the old domains. Also, the set of concepts that we want to detect in the target domain and the old domain are the same. These two methods work with the abstract feature vectors generated from the underlying data points. The prediction-based method, on the other hand, deals with the scenario where we have different types of data in the target and the old domains. Since the concept prediction scores generated from the old-domain models are used as features, the target concepts can be different from the old-domain concepts.

In terms of computation complexity, both CDSVM and AS^{3}VM are faster than the Combined SVM in general, especially with large-scale old domain data $\mathcal{X}^{old}$. This is because CDSVM only incorporates a part of the old data (previously learned support vectors), and AS^{3}VM relies on a matrix inversion instead of solving the QP problem. As for the prediction-based method, the major time complexity lies in the computation of the ELFs, including extraction of elementary-level features and construction of vocabularies. For example, to compute the region-level concept scores, we need to segment images and apply previous region-based concept detectors to the segmented regions. Luckily, a lot of computation can be conducted offline during the training process, and the evaluation is still reasonably fast.

1.8 Acknowledgment

This work was supported in part by a Kodak Fellowship from Eastman Kodak Company (to the first author) and NSF CRI Award No CNS-07-51078.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, (11):2399–2434, 2006.

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

[3] S.F. Chang, W. Jiang, W. Hsu, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Columbia University trecvid-2006 video search and high-level feature extraction. *NIST TRECVID workshop*, 2006. Gaithersburg, MD.

[4] O. Chapelle, B. Scholkopf, and A. Ziene. *Semi-supervised learning*. 2006. MIT Press, Cambridge, MA.

[5] H. Daumé. Frustratingly easy domain adaptation. *Proc. the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

[6] L. Duan, I.W. Tsang, and D. Xu. Domain transfer svm for video concept detection. *Proc. IEEE CVPR*, pages 1375–1381, 2009.

[7] S. Ebadollahi, L. Xie, S.F. Chang, and J. Smith. Visual event detection using multi-dimensional doncept dynamics. *Proc. IEEE ICME*, pages 881–884, 2006.

[8] W. Jiang and A.C. Loui. Semantic event detection for consumer photo and video collections. *Proc. IEEE ICME*, pages 313–316, 2008.

[9] W. Jiang and A.C. Loui. Effective semantic classification of consumer events for automatic content management. *The First SIGMM workshop on Social media*, pages 35–42, ©2009 ACM, Inc. http://doi.acm.org/10.1145/1631144.1631153

[10] W. Jiang, E. Zavesky, S.F. Chang, and A.C. Loui. Cross-domain learning methods for high-level visual concept classification. *Proc. IEEE ICIP*, pages 161–164, 2008.

[11] Y.G. Jiang, C. Ngo, and S.F. Chang. Semantic context transfer across heterogeneous sources for domain adaptive video search. *ACM Multimedia*, 2009. Beijing, China.

[12] A. Loui, J. Luo, S.F. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa. Kodak's consumer video benchmark data set: concept definition and annotation. *ACM SIGMM Int'l Workshop on MIR*, pages 245–254, 2007.

[13] A. Loui and A. Savakis. Automated event clustering and quality screening of consumer pictures for digital albuming. *IEEE Trans. on Multimedia*, (3):390–402, 2003.

[14] LSCOM lexicon definitions and annotations version 1.0. *DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia*, March 2006.

[15] M. Naphade, L. Kennedy, J. Kender, S.F. Chang, J. Smith, P. Over, and A. Hauptmann. A light scale concept ontology for multimedia understanding for trecvid 2005. *IBM Research Technical Report*, 2005.

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. *Proc. NIPS*, 2001.

[17] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, (2):99–121, 2000.

[18] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. *Proc. IEEE ICCV*, pages 1470–1477, 2003.

[19] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. pages 321–330, 2006.

[20] V. Vapnik. *Statistical learning theory*. Wiley-Interscience, New York, 1998.

[21] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. *Proc. ACM Multimedia*, pages 188–197, 2007.

[22] B. Yao, X. Yang, and S.C. Zhu. Introduction to a large scale general purpose ground truth data set: methodology, annotation tool and benchmarks. *Proc. IEEE Int'l Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2007.

[23] X. Zhu. Semi-supervised learning literature survey. 2005. Computer Sciences Technique Report 1530, University of Wisconsin-Madison.