Learning to Adapt Across Multimedia Domains

Jun Yang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
juny@cs.cmu.edu

November 19, 2007

Abstract

In multimedia, machine learning techniques are often applied to build models to map low-level feature vectors into semantic labels. As data such as images and videos come from a variety of domains (e.g., genres, sources) with different distributions, there is a benefit of adapting models trained from one domain to other domains in terms of improving performance and reducing computational and human cost. In this thesis, we focus on a generic adaptation setting in multimedia, where supervised classifiers trained from one or more *auxiliary domains* are adapted to a new classifier that works well on a *target domain* with limited labeled examples. Our main contribution is a *discriminative framework for function-level classifier adaptation* based on regularized loss minimization, which adapts classifiers of any type by modifying their decision functions in an efficient and principled way. Two adaptation algorithms derived from this general framework, adaptive support vector machines (aSVM) and adaptive kernel logistic regression (aKLR), are discussed in details. We further extend this framework by integrating *domain analysis* approaches that measure and weight the utility of auxiliary domains, and *sample selection* methods that identify informative examples to help the adaptation process. The proposed approaches are evaluated on cross-domain video concept detection using the TRECVID corpus, where preliminary experiments have shown promising results. Our general approaches can be applied to other adaptation problems including retrieval model adaptation and cross-corpus text categorization.

Thesis Committee:
Alexander G. Hauptmann (Chair)
Christos Faloutsos
Jie Yang
Shih-Fu Chang (Columbia University)

Contents

1 Introduction **4**

1.1 Motivation and Task Definition 4

1.2 Research Challenges 6

1.3 An Overview of our Approach 7

2 Literature Review **11**

2.1 Adaptation in Multimedia 11

2.2 Transfer and Multi-Task Learning, Incremental Learning, and Sample Bias Correction 12

2.2.1 Transfer learning 13

2.2.2 Multi-task learning 14

2.2.3 Incremental Learning 15

2.2.4 Sample Bias Correction 15

2.3 Concept Drift in Data Mining 16

2.4 Adaptation in Other Areas 17

3 Domain Impact on Video Concept Detection **18**

3.1 TRECVID Video Collections 18

3.2 Proposed Work: A Survey of Domain Impact on Video Concept Detection 19

3.3 Cross-Domain Video Concept Detection 19

4 A Discriminative Framework for Classifier Adaptation **20**

4.1 Problem Settings and Notations 20

4.2 A Discriminative Framework for Adaptation 20

4.3 Adaptive Support Vector Machines (aSVM) 22

4.3.1 Model Formulation 22

4.3.2 Discussion 24

4.3.3 Learning Algorithm of aSVM 25

4.4 Adaptive Kernel Logistic Regression (aKLR) 27

4.4.1 Model Formulation 27

4.4.2 A Probabilistic Interpretation 28

4.4.3 Learning Algorithm for aKLR 29

4.5 Preliminary Experiments 30

4.5.1 Approaches for Comparison 30

4.5.2 Performance: Synthetic Data 31

4.5.3 Performance: Cross-channel Video Concept Detection 32

5 Domain Analysis 38

5.1 Proposed Work: Adaptation with Domain Weighting 38

5.1.1 Extended Framework 38

5.1.2 Adaptive SVM with Domain Weighting (aSVM-DW) 39

5.1.3 Discussion of aSVM-DW 40

5.1.4 Learning Algorithm for aSVM-DW 41

5.2 Extraction of Domain Features 41

5.2.1 Domain Metadata 42

5.2.2 Features on Score Distribution 42

5.2.3 Pseudo Performance 44

5.2.4 Predicting Classifier Performance 44

5.2.5 Preliminary experiment 45

5.3 Proposed Work: Adaptation with Domain Features 46

6 Sample Selection for Domain Adaptation 47

6.1 Sample Selection based on Loss Minimization 47

6.1.1 The Theoretical Model 47

6.1.2 Computationally Tractable Methods 48

6.1.3 Batch and Incremental Sample Selection 50

6.1.4 Sample Selection with Multiple Auxiliary Classifiers 50

6.2 Proposed Work: Sample Selection based on Classifier Disagreement . . . 51

6.3 Preliminary Experiments on Sample Selection 51

7 Proposed Work: Domain Adaptation in Other Applications 53

8 Summary of Proposed Thesis Work 54

8.1 Domain Impact to Video Concept Detection 54

8.2 Domain Analysis 54

8.3 Sample Selection for Adaptation 54

8.4 Domain Adaptation in Other Areas 55

8.5 Timeline 55

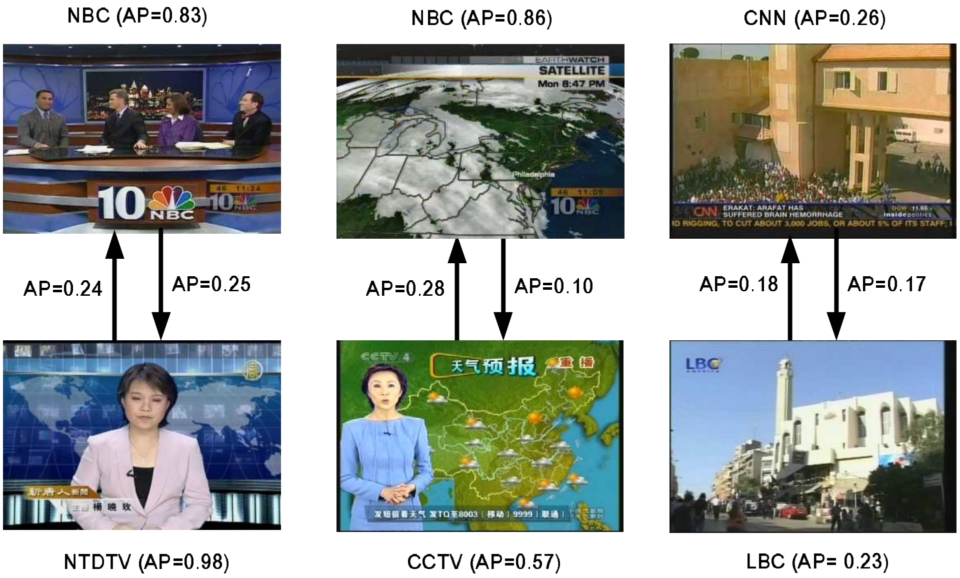
1 Introduction

The explosive growth of multimedia data makes their analysis, classification, and retrieval a critical problem in both research and industry. Recently, machine learning is playing an increasingly important role in this area, where models are built for tasks varying from classifying images into categories, detecting semantic concepts in video shots, to matching image and video data with user queries. With multimedia data coming from a variety of domains, such as different genres, producers, and media, there is a desire to generalize and adapt models trained from one domain to other domains. Compared with building new models for each domain, adapting existing models is beneficial in terms of computational and human resource and performance, but also challenging given that data distribution changes arbitrarily across domains. This thesis is dedicated to developing efficient and principled approaches to adapting models across multimedia domains. We will discuss the motivation, goal, and challenges of this research, and briefly overviews our approaches and the key contributions.

1.1 Motivation and Task Definition

Adapting models for multimedia data is necessary because the models have *poor generalizability* across different domains, i.e., models trained from data in one domain work well on this specific domain but poorly on other domains. This is caused by different data characteristics in different domains. More precisely, it is because the distribution of features representing multimedia data usually changes from one domain to another, and distribution mismatch is a fundamental problem that causes learning models to fail to generalize. An example is concept detection in news video, where models are built to recognize semantic concepts from video scenes. As shown in Figure 1, due to large visual discrepancy between video data from different news channels, the performance of a model trained from one channel drops substantially when applied to a different channel. For example, the "studio" scenes from NBC and NTDTV channel differ in terms of background, room setting, and the number of people, which causes performance of the model for NTDTV to plummet from 0.98 average precision on NTDTV to only 0.24 on NBC.

One can address the generalizability problem by building new models for every domain. However, we prefer adapting existing models over this costly approach because adaptation requires *fewer labeled examples* to achieve the same performance, or achieves *higher performance* using the same number of labeled examples. This is important given that labeling multimedia data is time-consuming and the size of training data needed to build reliable models is large. According to the statistics on TRECVID 2007 collaborative annotation [61], roughly 215 intense man-hours were spent on labeling 50-hour video w.r.t 36 concepts in order to build models for another 50-hour video. If this number does not appear daunting, imagine that this effort has to be repeated if one choose to build new models for every video collection. On the other hand, existing models trained from other domains provide valuable information to tasks in a new domain. This is shown by the "building" detector for LBC channel in Figure 1, which performs reasonably well even on the CNN data. Exploiting the knowledge in these out-of-domain models reduces the required labeled examples, and consequently, the human effort needed to label them and the cost for training models over them.



(a) "studio" scenes (b) "weather" scenes (c) "building" scenes

Figure 1: Example video scenes of three semantic concepts from different news video channels. We show the performance of concept detectors trained from each channel as measured by average precision (AP) when they are applied to the same channel and a different channel. There is a significant decline in performance when applying concept detectors to other channels.

Therefore, cross-domain model adaptation is necessary and beneficial in multimedia. However, this problem is so far *overlooked* or at best *understudied* in the multimedia community. Although there are techniques for adapting concept detectors across correlated concepts [55, 95, 60] or retrieval models across query classes [94, 45], no work is devoted to adaptation across data domains. Also, there has not been a general approach to various adaptation problems in multimedia based on a unified formulation.

Many learning tasks in multimedia can be formulated as *a classification problem* of mapping data description $\mathbf{x}$ to some semantic, categorical label y. For example, video concept detection is to map low-level features of video shots into a binary label indicating the presence (or absence) of a semantic concept. Other tasks such as image classification and multimedia retrieval can be formulated similarly. Therefore, *supervised classifiers* that learn such feature-label mappings $f : \mathbf{x} \rightarrow y$ are the most frequently used models for multimedia. The goal of this research is to develop efficient and principled approaches to adapting supervised classifiers across different domains of multimedia data. Formally, we define domain and model adaptation as:

Definition 1. *A **domain** is a set of multimedia data generated by the same data distribution $p(\mathbf{x})$ and class-conditional distribution $p(y|\mathbf{x})$. Concretely, a domain is described by image or video data belonging to a certain genre, or created by a specific producer, etc. For example, cartoon images and photographs are two image domains, news video and documentaries are two video domains, and news video from different channels can be also viewed as from different domains.*

Definition 2. ***Cross-domain model adaptation** is to adapt supervised classifiers for a given task trained from one or more **auxiliary domains** to a new classifier that works well on a different **target domain**. We call the existing classifiers **aux-***

***iliary classifiers** and the new classifier **target classifier**. There are two further assumptions: (1) The auxiliary domains are related to the target domain in the sense that auxiliary classifiers have better-than-random performance on the target domain; (2) Only a limited number of labeled data are available in the target domain, while the labeled data in the auxiliary domains are plenty.*

The first definition implies that domains may have *different distributions*. In some cases, the data distribution $p(\mathbf{x})$ changes across domains, while the class-conditional $p(y|\mathbf{x})$ stays the same. Nevertheless, classifiers trained from one domain are unlikely to capture the true $p(y|\mathbf{x})$ given the bias in data. For example, we apply an *"anchor"* classifier trained from CCVT news video, where anchors always appear in studios, to CNN news video, where anchors often appear outdoors. The definition of anchor does not change, which means $p(y|\mathbf{x})$ is the same, but the classifier cannot recognize outdoor anchors as it never see them before. In other cases, both $p(\mathbf{x})$ and $p(y|\mathbf{x})$ changes across domains. This happens when a retrieval model that combines various similarity scores into a relevance label is adapted across query classes, because, for example, the importance of face similarity score as part of $\mathbf{x}$ changes from person-related queries to other queries. Since both $p(\mathbf{x})$ and $p(y|\mathbf{x})$ may change, we make no assumption as to *whether* and *how* the distribution changes between auxiliary and target domain. This does not mean the two domains are totally irrelevant. Instead, we constrain their relation through classifier performance as specified by the second assumption in the above definition.

1.2 Research Challenges

Model adaptation has been studied in different areas and various related techniques have been proposed. In machine learning, inductive transfer and multi-task learning methods apply knowledge learned from one or more tasks to solving related tasks [23, 13, 50, 52, 88, 100]. In data mining, there has been research on recognizing drifting concepts from data streams [46, 76, 86]. Specialized adaptation methods are available to adapt language and parsing models in natural language processing (NLP) [5, 41, 65] and acoustic models in speech recognition [34, 49]. Nevertheless, the properties of multimedia data and their models raise several challenges which existing approaches are unable to fully address. The major challenges are:

- *Modeling distribution changes of multimedia data is technically infeasible.* As mentioned, data distribution $p(\mathbf{x})$ almost always changes across multimedia domains, and class-conditional $p(y|\mathbf{x})$ often changes. Changes of $p(\mathbf{x})$ are difficult to capture because there are no generic and accurate data models for multimedia, which can be represented by a variety of features. Modeling changes of $p(y|\mathbf{x})$ is simply impossible without sufficient labels in the target domain. Indeed, articulating distribution change is a harder problem than classifying the data, and solving it would make the solution to classification problems trivial. We thus prefer adaptation approaches that require no knowledge and make no assumptions on whether and how distribution changes. Existing methods [9, 10, 77, 99] relying on such knowledge or assumptions are not suitable.
- *Training on raw data from auxiliary domains is costly.* This is because models for multimedia data are expensive to train due to high-dimensional, real-valued

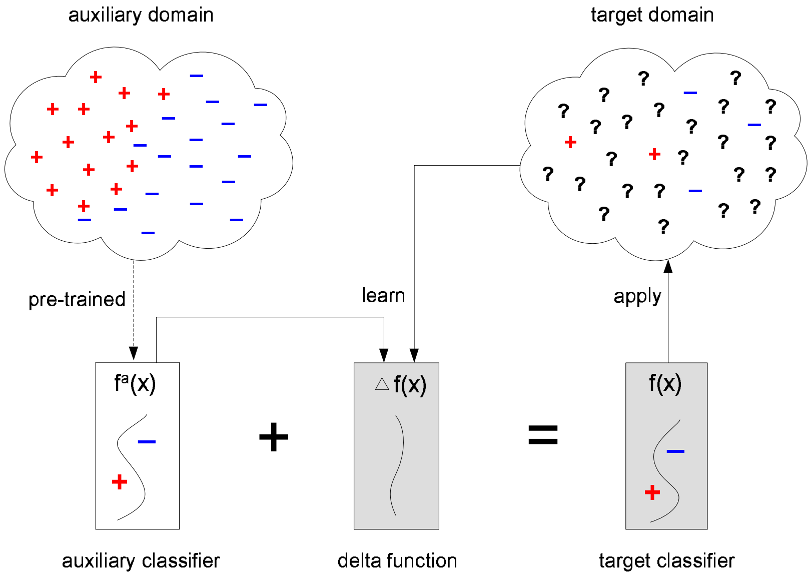


Figure 2: The basic framework for function-level model adaptation. The red '+' denotes positive instances, blue '-' denotes negative instances, and '?' denotes unlabeled instances. The shaded boxes denote components be learned, while the white boxes are existing components.

features and learning algorithms with superlinear cost w.r.t data size (e.g., SVM). Training the target classifier over data aggregated from all domains, a widely used approach in previous work [22, 46, 50, 88], is inefficient or intractable given the large size of auxiliary data. Auxiliary data can be also unavailable or inaccessible due to copyright or privacy issues. An efficient approach is desired to exploit more compact representations of domain knowledge than complete raw data.

- *Auxiliary domains are not equally useful.* Exploiting an unrelated auxiliary domain does more harm than good to solving problems in the target domain. Measuring the relatedness between auxiliary and target domain, and selecting and weighting the auxiliary classifiers accordingly, is critical to the success of model adaptation. This issue has not been addressed in previous work.
- *Examples in the target domain are not equally informative.* Since many classification problems in multimedia are *imbalanced*, examples from the rare class are more valuable. Also, in the context of adaptation, examples that provide *complementary* information to auxiliary domains are more useful. When we can label only a limited number of examples in the target domain, selecting more informative examples is particularly important.

1.3 An Overview of our Approach

To address these challenges, we organize our research around three correlated problems:

1. *How to adapt classifiers across domains in an efficient and principled way?*
2. *How to measure domain relatedness and the utility of auxiliary classifiers?*
3. *How to identify informative training examples to facilitate adaptation?*

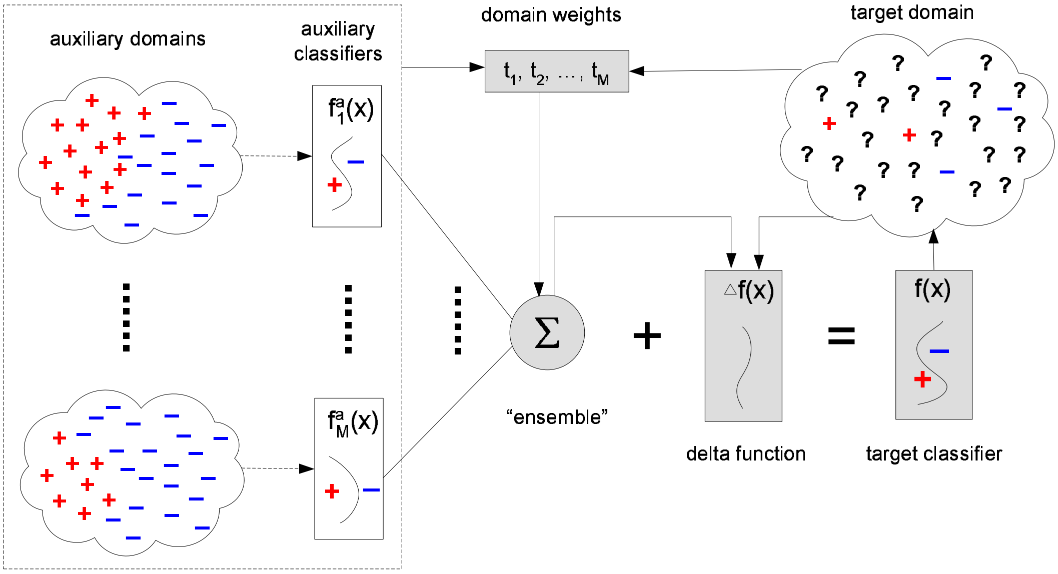


Figure 3: The extended adaptation framework with multiple auxiliary domains and domain analysis. The shaded components need to be learned.

The foundation of our work is a *discriminative framework for function-level classifier adaptation* based on loss minimization principle, which is illustrated in Figure 2. In this framework, an auxiliary classifier $f^a(\mathbf{x})$ of any type is adapted to a new classifier $f(\mathbf{x})$ for the target domain by adding a "delta function" $\Delta f(\mathbf{x})$ to the decision function of $f^a(\mathbf{x})$. The learning of $\Delta f(\mathbf{x})$ is based on the labeled examples in target domain and the auxiliary classifier $f^a(\mathbf{x})$, under dual objectives that the target classifier $f(\mathbf{x})$ achieves minimum classification error and its distance to $f^a(\mathbf{x})$ in the function space is also minimized. As a fundamental difference from extent approaches which directly use data from auxiliary domains, this framework exploits auxiliary classifier as a *summary* of the classification-related knowledge distilled from the raw data. Using such compact representation of domain knowledge implies high efficiency and broad applicability of our approach. In practice, this general framework can be "instantiated" into concrete adaptation algorithms by plugging in specific loss and regularization functions. We describe two such algorithms, namely *adaptive support vector machines (aSVM)* and *adaptive kernel logistic regression (aKLR)*, where the latter can be also derived from a probabilistic perspective. This addresses the first problem of model adaptation.

As shown in Figure 3, we extend the basic framework to accommodate multiple auxiliary domains and their weighting. In this framework, the target classifier $f(\mathbf{x})$ is obtained by adding the delta function $\Delta f(\mathbf{x})$ to a *weighted ensemble* of auxiliary classifiers $f^a_1(\mathbf{x}), ..., f^a_M(\mathbf{x})$, where the weights reflect the utility of these auxiliary classifiers w.r.t the target domain. We propose several *domain analysis* methods to measure the relatedness between auxiliary and target domains and determine the utility of auxiliary classifiers, first based on their performance on the labeled data in target domain and then based on domain features that indicate domain relatedness. We integrate these methods into our adaptation framework, leading to several extensions of aSVM such as aSVM with domain weighting (aSVM-DW) and aSVM with domain feature based weighting (aSVM-DFW). Moreover, we explore two *sample selection* methods for identifying informative examples from the target domain to facilitate adaptation. The first method selects examples to achieve the largest loss reduction on auxiliary classifiers, while the second one selects examples causing maximal disagreement between auxiliary classifiers. This addresses the second and third problem of model adaptation.

The proposed approaches are evaluated on the task of cross-domain video concept detection based on the TRECVID benchmark dataset [74]. Following a comprehensive survey on the impact of domain change to the performance of video concept classifiers, we demonstrate that our approaches can efficiently adapt concept classifiers to new domains with minimum human labeling effort. We also apply our approaches to other adaptation problems including adapting video retrieval models across queries, adapting text categorization models across corpora, and others to be determined.

We summarize the key advantages of our approach as follows:

- **Performance gain:** Based on the same set of labeled data, a classifier adapted from one in a related domain consistently outperforms a new classifier trained from scratch in terms of the quality of results. The gain is substantial when the labeled set is small, and does not fade out until a large amount of data is labeled.
- **"Black-box" applicability:** As a key benefit of function-level adaptation, our approach is able to adapt classifiers of *any type* by treating them as "black boxes", namely decision functions $f(\mathbf{x})$. This is important because a variety of classifiers are used in multimedia, including SVM, neural network, and so on. Being able to adapt without raw data in auxiliary domains also makes our approach applicable to tasks where such data are unavailable or inaccessible. This is critical for applications involving copyright-protected data or privacy-related data such as surveillance video.
- **High efficiency:** The freedom from training over auxiliary data results in high efficiency. The cost of adapting a classifier using limited target data is shown to be substantially lower than training one from all the data, because classifiers like SVM have super-linear training cost against data size and auxiliary data are typically plenty. Another aspect of efficiency comes from the ability to find the most informative examples, such that fewer examples need to be labeled to reach the same performance.
- **Maximal domain utility:** The domain analysis component makes educated decisions on weighting the utility of auxiliary domains (classifiers). It therefore maximizes the overall utility of auxiliary domains.
- **Extendability:** The proposed framework advocates a *methodology* for constructing adaptation algorithms. One can drive virtually an infinite number of algorithms from this framework by varying the loss and regularization function.

The remainder of the proposal is organized as follows. Section 2 reviews related work in multimedia, machine learning, data mining, and other areas. Section 3 surveys the impact of domain change to video concept detection, which is the benchmark task of the proposed approach. Section 4 describes the discriminative framework for function-level classifier adaptation and the aSVM and aKLR algorithm derived from this framework. Section 5 proposes domain analysis approaches for measuring domain relatedness and utility of auxiliary classifiers, and Section 6 proposes two sample selection methods in the context of adaptation. Section 7 explores the application of the proposed approaches in various adaptation tasks other than the benchmark task. Finally, Section 8 summarizes the proposed work and gives a tentative timeline.

2 Literature Review

In this section, we review previous work related to cross-domain model adaptation in different research areas. We first discuss the approaches for adaptation problems in multimedia, which is the main application of this thesis. This is followed by a review of related works on transfer learning and multi-task learning in the machine learning area, concept drift detection in the data mining area, and adaptation in specific areas such as speech recognition and natural language processing. We will focus on the connections and differences between the existing techniques and our approach.

2.1 Adaptation in Multimedia

Recent work on the analysis, classification, and retrieval of multimedia data involve building various machine learning models, particularly supervised classifiers. In image classification and video concept detection [1, 16, 38, 60, 74, 75, 79, 85, 89, 95], supervised classifiers are used to map features representing images or video shots into labels indicating the categories or concepts they belong to. These classifiers are trained from manually labeled data using learning methods such as support vector machines (SVM) and neural network. Moreover, multimedia retrieval models [1, 45, 87, 94, 92] need to combine the relevance scores computed from multiple knowledge sources, such as keyword similarity and image similarity. The combination weights can be estimated using classifiers such as logistic regression trained from past queries and their (labeled) results [45, 94, 92]. As suggested by Natesv et al. [57], classification and retrieval can be viewed as the same problem of classifying multimedia data as relevant or irrelevant w.r.t categories or queries. While supervised classifiers are perhaps the most widely used learning models in multimedia, there are many other types of models, such as models for annotating images with keywords [7, 11, 36, 42], for labeling video shots with person names [68, 97] and with locations [96].

In terms of performance and (computational and human) efficiency, there is a need and benefits for adapting models of multimedia data across different domains. However, the problem of model adaptation is in general overlooked or at best understudied in multimedia. Most work in multimedia assume the training and test data come from the same domain. For example, image classification and annotation models [7, 16, 32] are often trained and tested on selected subsets of the Corel database, which contains high-quality photos. Most video concept detection and video search models [1, 38, 57, 60, 75] are trained and tested on the TRECVID collection [74] of each year (from 2002 to 2007), which contains a specific type of video such as documentary and broadcast news video. While there is nothing wrong with this evaluation setting, in practice we may have to work with data from different domains simultaneously or one by one. The need of adaptation becomes apparent if we want to reuse existing models while shifting from one domain to another.

Previous work on two specific problems in multimedia can be related to model adaptation, although the proposed methods are specialized to those two problems. One of them is video concept detection based on the correlations between multiple semantic concepts. Methods for this problem learn classifiers of multiple concepts simultaneously, which perform better than classifiers learned independently for individual concepts. For example, Naphade et al. [55] explicitly modeled the linkages between concepts via a Bayesian network that implicitly offered ontology semantics in a video collection. Amir et al. [1] concatenated the prediction scores of various correlated concepts into a long feature vector called “model vector”, based on which a SVM classifier was built for each concept. Yan et al. [95] proposed a series of probabilistic graphical models to mine the relationships between concepts. In addition, Qi et al. [60] proposed correlative multi-label (CML) framework and Chen and Hauptmann [18] proposed multi-concept discriminative random field (MDRF) to automatically identify concept correlations and learn concept classifiers simultaneously. These two methods are similar in spirit as they modify the regularization term of support vector machines (in [60]) or logistic regression (in [18]) to accommodate the correlations between different concepts. This problem has also been studied by Snoek et al. [75] and by Wu et al. [91]. However, these methods do not adapt concept classifiers of one concept to another. There is also no prior work exploring the correlations between data domains for the purpose of concept detection.

Another related research is on adapting multimedia retrieval models towards new queries with or without users’ feedbacks on retrieval results. Relevance feedback methods for content-based image retrieval [20, 67] update the initial query representations, distance metrics in the feature space, and/or retrieval models based on user feedbacks in order to improve retrieval results. In video retrieval, Yan [92] proposed to construct the retrieval model for a new query as a mixture of existing models of several predefined “query classes”, and update the model based on users’ implicit and explicit feedbacks to better reflect the characteristics of the new query. This work is extended by Kennedy et al. [45] to allow query classes to be automatically discovered. These methods are designed for specific feature representation and retrieval algorithms and not applicable to general adaptation problems.

To summarize, the issue of model adaptation has been studied for some *specific* problems in multimedia, yet there lacks a *generic* and *systematic* approach. In this thesis, we take an unified view of various adaptation problems in multimedia, as well as a generic and systematic approach to these problems. Specifically, we will focus on adapting *supervised classifiers* since they are arguably the most widely used model in multimedia. A generic approach avoids the need of developing algorithms for different problems, and allows people to treat emerging adaptation problems with ease.

2.2 Transfer and Multi-Task Learning, Incremental Learning, and Sample Bias Correction

In machine learning, research along several directions are relevant to the problem of cross-domain model adaptation. Transfer learning and multi-task learning both refer to the notion of applying knowledge learned from one or more tasks to related tasks, where the former refers to the transfer from related tasks to a target task, while the latter refers to the case of learning multiple related tasks together. Cross-domain adaptation can be viewed as transfer learning between multiple data sets on the same task, and our approach can be treated as a function-level transfer learning approach. Besides, incremental learning continuously updates a model based on subsets of the training data, and sample bias correction deals with learning problems where the distribution of test and training data are different. We review previous work in these directions and discuss their relation to our problem.

2.2.1 Transfer learning

Transfer learning (TL) aims to apply knowledge learned from *auxiliary tasks*, where labeled data are usually plenty, to develop an effective model for a related *target task* with limited labeled data. There is no formal definition of "related tasks", and in practice it refers to either related learning problems on the same dataset or the same learning problem on different datasets. After the notion was first introduced by Thrun [81] about 10 years ago, approaches have been proposed to transfer the knowledge at different levels of abstraction, including data level, representation level, and parametric level.

Transfer at *data level* augments the training data of the target task with labeled data from auxiliary tasks in order to build a better model for the target task. This has been the spirit in TL approach for k-nearest neighbor [81], for support vector machines by Wu and Dietterich [88], for logistic regression by Liao et al. [50], and for AdaBoost by Dai et al. [22]. While some of these methods (e.g., [88, 22]) do not directly add auxiliary data into the training set, a close examination reveals that such data play the role of additional training data in these methods. A key issue is the weights of data from the auxiliary tasks, which can be specified manually [88] or according to the degree of "mismatch" with the target data [50, 22]. Efficiency is the main disadvantage of these methods, which training can be considerably more expensive due to the large number of auxiliary data.

Approaches at *representation level* learn an effective feature representation and/or distance metric from the auxiliary tasks and use it for the target task. In fact, the first work on transfer learning [81] suggested to learn both a new data representation and a distance function from the labeled data in auxiliary tasks. Moreover, Ben-David et al. [23] derived a common representation that makes the auxiliary and target (data) domains appear to have similar distributions in order to enable effective adaptation, and Raina [62] used unlabeled images collected from various sources to learn high-level feature representations that can make image classification tasks easier in general. These approaches are obviously more general and efficient than data-level transfer learning methods, since the representation needs to be learned only once and is applicable to many other tasks.

Parameter-level approaches use the parameters of previously learned models from related tasks to form a "prior" for the model parameters to be learned for the target task. The new model can be thought as a "posterior" obtained by updating the prior in the light of the examples in the target task. Many approaches choose Bayesian logistic regression with a Gaussian prior (on parameters) as the learning algorithms for the primary task. In Marx [52], both the mean and variance of the prior are computed from the parameters of the models for related tasks. Similarly, Raina et al. [63] constructed a Gaussian prior with its covariance matrix encoding the word correlations derived from text classification tasks and applied it to similar tasks. Zhang [101] combined Rocchio algorithm with logistic regression via a Gaussian prior to yield a low-variance model for adaptive filtering. Fei-fei [31] implemented the Bayesian prior in more sophisticated models for one-shot learning of object categories.

There are also more specialized TL approaches. For instance, Taylor and Stone [80] proposed a transfer algorithm for reinforcement learning, and Heitz et al. [39] described a landmark-based model for transfer the "essence" of object classes learned from cartoon images to natural images. The issue of transfer with multiple auxiliary tasks was discussed by Marx et al. [52] and a simple solution was provided. Rosenstein et al. [66] explored the impact of the relatedness between target and auxiliary tasks on the performance of transfer learning.

The framework to be proposed in this thesis can be viewed as transfer learning at a even higher level: the *function level*. Our framework aims to directly adapt the decision functions of one or more auxiliary classifiers into the decision function of a classifier for the target data. Function-level transfer learning offers many benefits, including high efficiency by avoiding training over auxiliary data and flexibility from the freedom of using auxiliary classifiers of any types. To our knowledge, the most similar work in the literature has been that by Schapire et al. [69], which modifies the AdaBoost algorithm such that the Kullback-Leibler divergence between the classifier learned and a prior model representing human knowledge is minimized. However, this method is specialized for AdaBoost, our framework is able to adapt classifiers of any type. Moreover, issues such as weighting of auxiliary classifiers and sample selection will be discussed in the same framework.

2.2.2 Multi-task learning

Multi-task learning (MTL) explores the dependency between related tasks, and aims to achieve better performance than learning each task independently and to generalize previously learned knowledge for benefiting new tasks in future. Unlike in TL where the knowledge transfer is unidirectional (from auxiliary tasks to target task), in MTL the knowledge transfer is be mutual and between any related tasks. Despite the difference on problem setting, MTL approaches provide important references to our cross-domain adaptation approach as they both deal with knowledge transfer between tasks.

Similar to the approaches for TL, most MTL approaches support the transfer of knowledge either at the representation level or at the parametric level. *Representation-level* approaches [2, 3, 4, 13, 100] learn to map features into a latent feature space (i.e., representation) shared by all the tasks, and simultaneously learn the model for each task independently in this common feature space. The shared representations are derived to achieve maximal independence through latent independent component analysis in Zhang et al. [100], to maximize sparsity (i.e., lower dimension) in the method by Argyriou and Evgeniou [4], or under other heuristics. A related method is that by Niculescu-Mizil and Caruana [58], which learn the structure of a Bayes net from related tasks. *Parametric-level* approaches assume the model parameters of multiple related tasks must be related in a certain way and can be optimized together. The constraint on the model parameters is realized by a common Bayesian prior on all the model parameters as in the method of Bakker and Heskes [6], or by a regularization term penalizing the distance between model parameters in the work of Evgeniou et al. [28, 27]. The work in [98] extended the idea using a hierarchical Bayesian method, where the models of related tasks in the form of Gaussian processes (GPs) are constrained in the function space, i.e., characterized by a common mean function and kernel. In addition, Lawrence and Platt [48] studied sample selection strategies and Ben-David and Schuller [8] provided a mathematical notion of task relatedness.

2.2.3 Incremental Learning

Incremental learning, or online learning, is to continuously update a model based on subsets of the data. It is preferred over training a model in one batch using all the data when the latter is computationally intractable due to large data size or when new data become available after the model is trained. There are many incremental learning methods for updating classifiers especially SVMs. For example, Syed et al. [78] proposed to update a SVM model by re-training it from the previous support vectors and the new data, and Cauwenberghs and Poggio [14] extended it by allowing also decremental learning of SVM.

On the surface, our adaptation approach is similar to incremental learning methods because both update existing classifiers based on additional labeled examples. Nevertheless, there are fundamental difference between them in terms of assumption, goal, and approach. Incremental learning approaches assumes the same underlying distribution, and their goal is that the incrementally trained models are identical or close approximation of models trained in one batch. In comparison, our adaptation approach acknowledges likely distribution changes across domains, and it does not require the adapted models to be close to batch-learned models (and they should not be). In fact, a major contribution of this thesis is on analyzing the relatedness between domains and weighting their contribution accordingly. Moreover, most incremental learning methods involve at least part of old training data (such as support vectors) in the training. Hence, they are not as efficient as our approach which directly adapts the existing classifier and uses absolutely *no* old data. To summarize, our adaptation approach can be used for efficient incremental learning, but not the other way around.

2.2.4 Sample Bias Correction

Another line of research focuses on learning with distinct training and testing distributions, a problem known as *sample selection bias* or *covariance shift*. In this problem setting, the training sample is govern from an unknown distribution $p(\mathbf{x}|\lambda)$ while the unlabeled test data is governed by a different unknown distribution $p(\mathbf{x}|\theta)$. The training and test distribution may differ arbitrarily, but there is only one true unknown conditional distribution $p(y|\mathbf{x})$. The goal is to find a classifier $f : \mathbf{x} \longmapsto y$ that can accurately classify the test data.

Many sample bias correction methods are based on the theorem that the expected loss on test distribution $p(\mathbf{x}, y|\theta)$ equals the *weighted* expected loss on the training distribution as $p(\mathbf{x}, y|\lambda)\frac{p(\mathbf{x}|\theta)}{p(\mathbf{x}|\lambda)}$. This means one can train a classifier that minimizes the loss on the test distribution by weighting the training loss with an instance-specific scaling factor $\frac{p(\mathbf{x}|\theta)}{p(\mathbf{x}|\lambda)}$. Since $p(\mathbf{x}|\lambda)$ and $p(\mathbf{x}|\theta)$ are typically unknown, Shimodaira [73] and Sugiyama et al. [77] propose to estimate $\hat{p}(\mathbf{x}|\lambda)$ and $\hat{p}(\mathbf{x}|\theta)$ from the training and test data using kernel density estimation, and then use $\frac{\hat{p}(\mathbf{x}|\theta)}{\hat{p}(\mathbf{x}|\lambda)}$ to resample or weight the training examples in the training of classifiers. Instead of estimating the data distribution $p(\mathbf{x}|\lambda)$ and $p(\mathbf{x}|\theta)$, Zadronzy [99] and Bickel and Scheffer [10] directly estimate the ratio $p(s = 1|\mathbf{x}, \lambda, \theta) \propto \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x}|\lambda)}$, where s is a selector variable that decides whether an example $\mathbf{x}$ drawn under the test distribution $p(\mathbf{x}|\theta)$ is moved into the training set ($s = 1$) or not ($s = 0$). In practice, $p(s = 1|\mathbf{x}, \lambda, \theta)$ is estimated from the training and test data, usually by a discriminative approach [10]. Besides, Huang et al. [40] devised the kernel mean matching method that sets the weights of training instances such that the first moment of training and test data matches. Bickel et al. [9] improved upon these methods by integrating the weight estimation and model training into a unified discriminative framework. Fan [30] suggested to use simple model averaging technique to alleviate the effect of sample bias.

Despite the advance on sample bias correction methods, several difficulties limit their applicability on our problem. One of them is the difficulty of estimating the distribution of multimedia data $p(\mathbf{x})$, not to mention its change across domains. While several generative models for image and video data exist [11, 90], they are specialized for certain features (e.g., color histogram) and have not shown to be effective. Moreover, the class-conditional $p(y|\mathbf{x})$ often changes. This violates the basic assumption of sample bias correction and renders all the aforementioned methods inapplicable. Thus, in this thesis we try to adapt models *without* articulating how the data distribution changes across domains.

2.3 Concept Drift in Data Mining

Domain adaptation is also related to the problem of concept drift in the mining of streaming data. This refers to the case where statistical properties of a target variable or *concept*, which a model is trying to predict, changes over time due to the change of some hidden context. An example of concept drift is spam detection from a user's daily emails, where the definition of spam may change over time (e.g., to include email advertisements) and the types of spam can change drastically with time. Concept drift may involve changes of the target concept definition (i.e., $p(y|\mathbf{x})$) and/or the change of the data distribution (i.e., $p(\mathbf{x})$).

There are two major approaches in data mining to detecting a drifting concept in time-evolving data. The first approach selects training instances using a temporal window with fixed or adaptive size that moves over the data stream, possibly weights the selected instances by their utility to the target concept, and uses them to build a single classifier. This has been the approach employed by Klinkenberg and Joachims [46], and by Cunningham et al. [21]. The second type of approaches maintains an (weighted) ensemble classifier that combines the outputs of a set of base classifiers learned from selected chunks of the data stream. Wang et al. [86], Street and Kim [76], and Kolter and Maloof [47] adopted this approach.

Despite being proposed for streaming data, the aforementioned methods can be adapted to solve domain adaptation problems if we treat each domain as a data chunk from a data stream. However, methods for concept drift usually make implicit assumptions about the underlying distribution of the data stream. For example, many assume [86, 46] the distribution of current data is similar to that of the most recent data, so that the model can be trained mainly on the recent data, and in the ensemble approach the base classifiers can be selected and weighted based on their performance on recent data. While such assumption may be reasonable in temporally correlated streaming data, no similar assumptions can be made on the distributions of domains, which are allowed to differ arbitrarily. In fact, as discussed in Section 1, measuring the relatedness between different domains is one of the key challenges to be tackled in this thesis. A related work from data mining is from Fan [29], which analyzed the impact of combining "old data" for training and proposed an efficient and systematic way to selecting useful data.

2.4 Adaptation in Other Areas

Model adaptation has been also studied in specific problem domains in natural language processing, speech, and many other domains. In natural language processing, there is often a need to adapt language models, parsers, and models for various tasks such as named entity detection, from one corpus to other corpora. Through experiments on Wall Street Journal corpus and Brown corpus, Gildea [35] found that statistical parsing models, especially the bigram statistics, are very corpus-specific, and suggested a technique to pruning model parameters to achieve better generalization ability. Hwa [41] proposed a two-stage adaptation process to first train a grammar from a fully-labeled old domain and re-estimate the probabilities of the grammars from a sparsely-labeled new domain. Roark and Bacchiani [65] investigated adapting a lexicalized probabilistic context-free grammar (PCFG) from an old domain to a new domain. Their approach is to compute the maximum a posterior (MAP) estimation of the model parameters under a prior distribution given by the old ("out-of-domain") models. In [5], they applied the same approach to adapt language models across corpora, where a language model is a generative model governing the distribution of terms and phrases in documents. In addition, Shen et al. [72] explored adapting a general hidden Markov Model (HMM) named entity recognizer from newswire domains to a biomedical domain. While most of the aforementioned models are generative ones, Chelba and Acero [17] proposed to adapt maximum entropy classifiers for recovering correct capitalizations in uniformly cased text based on MAP estimation of model parameters.

Model adaptation has been extensively studied in speech recognition, where the model mismatch problem can be caused by different speakers, dialects, speaking styles, and environments (noise levels). Since the acoustic model of a speech recognition system is usually based on hidden Markov model (HMM), most adaptation methods learn to adjust the parameters of the HMM model to better fit the target data. Such methods include speaker adaptation methods based on Maximum-a-posterior (MAP) [34], Maximum Likelihood Linear Regression (MLLR) [49], Vocal Tract Length Normalization (VTLN) [84], and noise adaptation methods such as parallel model combination (PMC) [33]. Overall, these adaptation methods are specialized to the (generative) acoustic models used in speech recognition systems.

3 Domain Impact on Video Concept Detection

Video concept detection is a typical learning problem in multimedia whose performance is vulnerable to the change of data domains. The goal of video concept detection is to automatically determine whether certain semantic concepts (e.g., *Studio*, *Outdoor*, and *Sports*) are present in video shots, where a video shot is a sequence of video frames taken in a single camera operation. The standard approach to this task is to build supervised *concept classifiers* which map the low-level features of video shots into binary labels indicating the presence (or absence) of the given concepts.

Conventionally, concept classifiers are trained and evaluated from the video data from a single domain, such as broadcast news video of certain channel(s). There is no study on the performance of these concept classifiers when being applied to domains other than their training domain, although such scenario is likely in practice. In this section, we conduct a comprehensive survey on the impact of domain change to the performance of video concept detection, in order to provide insights on the importance and challenges of the model adaptation problem in multimedia. We also describe cross-domain video concept detection as a benchmark task against which the proposed approaches will be evaluated.

3.1 TRECVID Video Collections

An ideal corpus for studying the impact of domain change to video concept detection is a large, heterogenous corpus consisting of data from multiple domains. A good choice is the video collections used in TREC Video Retrieval Evaluation (TRECVID) [74]. TRECVID is an annual workshop sponsored by the National Institute of Standards and Technologies (NIST) to promote research in content-based video retrieval in large collections via an open, metrics-based evaluation. It has defined a set of retrieval-related tasks for evaluation, including shot boundary detection, high-level feature extraction (a.k.a video concept detection), and automatic and interactive search. From 2002 to 2007, the TRECVID collection varies on a yearly basis from documentaries and movies to broadcast news video and documentary video. Among all the TRECVID collections, we select the "development set" of the collection used in 2005 and in 2007, which are referred to as TRECVID 2005 and TRECVID 2007 in this thesis.

TRECVID 2005 collection contains broadcast news video footage of 86 hours in length. The footage belongs to 6 different TV channels, including CNN, NBC, MSNBC, CCTV, NTDTV, and LBC. Among them, CCTV and NTDTV are in Chinese (Mandarin), LBC is in Arabic, while the others are in English. The data in each channel come from 2-3 different news programs, e.g., the footage from CNN are from "Live From CNN" and "Anderson Cooper 360". Due to the difference on editing styles, target audience, and other factors, the data from different channels exhibit different characteristics. The 86-hour footage has been manually partitioned into 61,901 video shots and the shot boundaries are provided. The footage is relatively evenly distributed across different channels, with "largest" channel containing 11,025 shots and the "smallest" one having 6,481 shots. Each shot is represented by one video frame as its "keyframe", usually the one in the middle of its duration. Each keyframe is depicted by a 273-d feature vector, which consists of a 225-d color moment feature computed from 5×5 grids and a 48-d Gabor texture feature. Since there is not much change in the content within a shot, we use this 273-d keyframe feature as the feature of the video shot.

As part of the Light Scale Concept Ontology for Multimedia (LSCOM-Lite) project [54], all the shots in TRECVID 2005 have been manually annotated with (binary) labels indicating the presence or absence of 39 semantic concepts. These concepts cover a wide variety of types, varying from outdoor scene (*Building*, *Road*), indoor setting (*Studio*, *Meeting*), to news genre (*Sports*, *Entertainment*), and general objects (*Airplane*, *Animal*).

3.2 Proposed Work: A Survey of Domain Impact on Video Concept Detection

We plan to conduct an comprehensive study on the impact of domain change to the performance of video concept detection based on TRECVID video collections. This is realized by building concept classifiers from one domain and comparing their performance on the data from the same domain (i.e., in-domain data) and the data from other domains (i.e., out-of-domain data). Specifically, this experiment will be performed in two settings: cross-channel and cross-genre setting. In the cross-channel setting, we evaluate the concept classifiers trained from one channel (of broadcast news footage) applied to the other channels, based on the multi-channel TRECVID 2005 collection. In the cross-genre setting, we evaluate the concept classifiers trained from broadcast news video in TRECVID 2005 collection applied to documentary video in TRECVID 2007 collection. This study not only compares the in-domain and out-of-domain performance of concept classifiers, but also examines how the performance change is affected by factors including the relatedness between domains, the types of semantic concepts, and so on.

3.3 Cross-Domain Video Concept Detection

It is clear from the survey that concept classifiers generalize poorly to domains other than its training domain. They need to be *adapted* to other domains. We call the task of detecting video concepts in one domain using concept classifiers that are adapted from existing concept classifiers built for other domains as *cross-domain video concept detection*. We use this task as the *benchmark task* against which we evaluate our adaptation approaches. This is because this task represents a set of common challenges faced by other adaptation problems in multimedia: Concept classifiers are vulnerable to the change of underlying data distributions across domains; The distribution change is arbitrary and hard to model; Training concept classifiers are computationally expensive, and labeling the training data is tedious and time-consuming. It is reasonable to assume that our approach is applicable to various adaptation problems in multimedia if it works well on cross-domain video concept detection.

4 A Discriminative Framework for Classifier Adaptation

As the foundation of the propose work, we propose a discriminative framework for function-level classifier adaptation, as well as two adaptation algorithms derived from the framework, namely adaptive support vector machines (aSVM) and adaptive kernel logistic regression (aKLR).

4.1 Problem Settings and Notations

We begin by defining a general problem setting and introducing the terminologies and notations used in this thesis. We consider a binary classification task in a *target domain*, where only a limited set of data are labeled while most data are unlabeled. We denote the labeled data as $\mathcal{D}^t = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where N is the number of instances, $\mathbf{x}_i$ is the feature vector of the i_{th} instance, and $y_i \in \{-1, +1\}$ is its binary label indicating relevance (or irrelevance) to a concept or category. For notational simplicity, we let each data vector $\mathbf{x}$ always include a constant 1 as its first element such that $\mathbf{x}_i \in \mathbb{R}^{d+1}$, where d is the number of features.

In addition to the target domain, there is an *auxiliary domain* which contains a large set of data $\mathcal{D}^a$ labeled w.r.t the same concept or category. Similarly, we have $\mathcal{D}^a = \{(\mathbf{x}_i^a, y_i^a)\}_{i=1}^{N^a}$, where N^a is the number of instances, $\mathbf{x}_i^a \in \mathbb{R}^{d+1}$ and $y_i^a \in \{-1, +1\}$. The distribution of auxiliary data $\mathcal{D}^a$ may be relevant but different from the distribution of target data $\mathcal{D}^t$ in an unknown way. A binary classifier has been trained from the auxiliary data $\mathcal{D}^a$ and is denoted as *auxiliary classifier* $f^a(\mathbf{x})$. This classifier can be trained using *any* classification algorithms (e.g., SVM, decision tree), but it is subject to a uniform representation as a decision function that predicts the data label as its sign, i.e., $\hat{y} = sgn(f^a(\mathbf{x}))$. For simplicity, we do not distinguish classifier and the decision function of classifier.

Sometimes, there are multiple auxiliary domains and their respective labeled data sets are denoted as $\mathcal{D}_1^a, ..., \mathcal{D}_M^a$. We define $\mathcal{D}_k^a = \{(\mathbf{x}_i^k, y_i^k)\}_{i=1}^{N_k^a}$, where $\mathbf{x}_i^k \in \mathbb{R}^{d+1}$ and $y_i^k \in \{-1, +1\}$. The distribution of these auxiliary datasets can be different from that of target data in different ways. We have trained an auxiliary classifier $f_k^a(\mathbf{x})$ from each auxiliary dataset $\mathcal{D}_k^a$.

The goal of our research is to propose a general framework for adapting an auxiliary classifier $f^a(\mathbf{x})$, or a set of auxiliary classifiers $f_1^a(\mathbf{x}), ..., f_M^a(\mathbf{x})$, to a *target classifier* $f(\mathbf{x})$ that works well on the target domain, based on the limited number of labeled examples $\mathcal{D}^t$ in the target domain. As a fundamental difference from many existing methods, our adaptation approach directly updates the decision function of the classifier. It does not use any auxiliary data or require them to be available.

4.2 A Discriminative Framework for Adaptation

We propose a generic and principled discriminative framework for *function-level classifier adaptation* based on regularized loss minimization principle. In this framework, the target classifier $f(\mathbf{x})$ has an *additive form*: it is the sum of the auxiliary classifier

$f^a(\mathbf{x})$ and a delta function $\Delta f(\mathbf{x})$:

$$f(\mathbf{x}) = f^a(\mathbf{x}) + \Delta f(\mathbf{x}) \tag{1}$$

This means that the transition from $f^a(\mathbf{x})$ to $f(\mathbf{x})$ is realized by adding a small function $\Delta f(\mathbf{x})$ on top of $f^a(\mathbf{x})$. The delta function $\Delta f(\mathbf{x})$ is learned from the labeled examples $\mathcal{D}^t = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ in the target domain under the influence of auxiliary classifier $f^a(\mathbf{x})$. We propose to learn $\Delta f(\mathbf{x})$ in a framework that aims to minimize the regularized empirical risk [37]:

$$\min_{\Delta f} \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda \Omega(\|\Delta f\|_{\mathcal{H}}) \tag{2}$$

where L is the empirical loss function, $\Omega(\cdot)$ is some monotonically increasing regularization function on the domain $[0, +\infty]$, $\|\cdot\|_{\mathcal{H}}$ is the norm of a function in a reproducing kernel Hilbert space (called RKHS) $\mathcal{H}$ as a function space, and λ is a scalar.

In Eq.(2), the first term measures the classification error (loss) of the target classifier $f(\mathbf{x})$ on the training examples; The second term is a regularizer that controls the complexity of the hypothesis space. Because $\|\Delta f\|_{\mathcal{H}} = \|f - f^a\|_{\mathcal{H}}$, this regularizer measures the distance between the auxiliary and target classifier in the function space. Hence, the target classifier $f(\mathbf{x})$ learned under this framework must satisfy two goals:

1. *minimal classification error on the training examples;*
2. *minimal distance from the auxiliary classifier $f^a(\cdot)$.*

While the second goal does not seem to be as intuitive as the first one, it is as important. If minimal classification error is the only goal, one may find a large number of classifiers achieving the same classification error (even zero classification error when the training size is small), although many of them do not generalize well beyond the training examples. The regularizer in Eq.(2) uses the distance to the auxiliary classifier as a second criterion for ranking candidate classifiers. This can be justified by our assumption that the auxiliary classifier has better-than-random performance on the target domain. The two goals is balanced by constant λ, and in practice its value needs to be determined based on the utility of the auxiliary classifier.

We can understand this adaptation approach as making *minimum* changes to the auxiliary classifier that are *necessary* to correctly classify the labeled examples. This *"minimum necessary changes"* principle is what underlies our adaptation framework. This is illustrated in Figure 4. The classification boundary A is trained from the auxiliary domain, and its performance on the target domain is suboptimal due to the distribution change. Our adaptation approach tries to find a new decision boundary B which is slightly modified from A but can classify the target data well.

In terms of bias-variance tradeoff, this framework attempts to reduce the high variance caused by limited training examples using the auxiliary classifier trained from sufficient out-of-domain data. It represents a middle way between two extremes, namely using a unbiased, high-variance classifier trained exclusively from limited examples, and using the low-variance but probably-biased auxiliary classifier. We expect the adapted classifier achieves better bias-variance tradeoff.

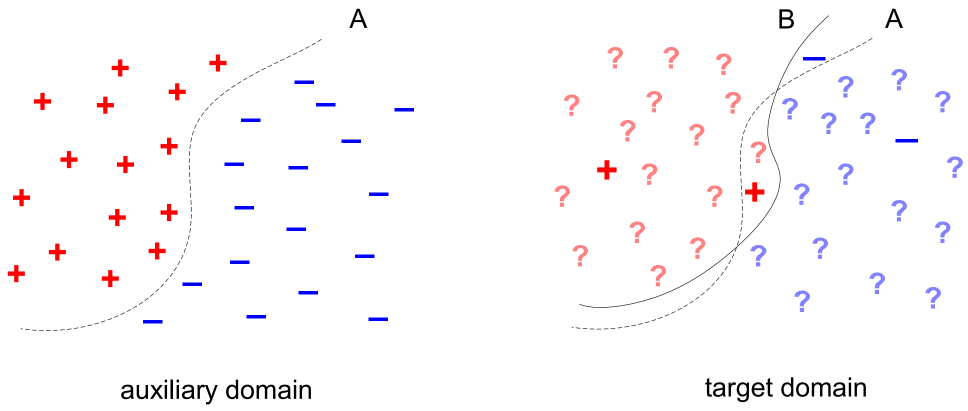


Figure 4: An illustration of classifier adaptation, where red '+' denotes positive instances, blue '-' denotes negatives, and red and blue '?' denotes unlabeled positive and negative instances. The decision boundary A is trained from the labeled data in the auxiliary domain, B is trained from the labeled data in the target domain, and C is adapted from A.

Taking a function-level adaptation approach results in great flexibility and efficiency. This is because our framework directly exploits the auxiliary classifiers as a *summary* of the knowledge of auxiliary domains, instead of using the raw data in auxiliary domains. By using this compact representation of knowledge, our approach is more efficient than existing adaptation methods that train models over typically a large number of auxiliary data [22, 50, 88]. On the other hand, this framework is applicable even when the auxiliary data are not available or accessible, which is usually the case in applications involving copyright-protected or privacy-related data such as surveillance video.

From this generic framework, one can derive concrete algorithms for classifier adaptation by choosing certain loss functions $L(\cdot)$, regularization functions $\Omega(\cdot)$, and the form of the delta function $\Delta f(\cdot)$. While the choices are virtually infinite, we will focus on two specific algorithms coming out of this framework, which adopt the loss function of support vector machines (SVM) and of kernel logistic regression (KLR) respectively. The first takes the advantage of the discriminative power of SVM, while the second gives a probabilistic interpretation of our framework.

4.3 Adaptive Support Vector Machines (aSVM)

The empirical success of support vector machines (SVM) in various classification problems has demonstrated the effectiveness of its hinge loss function. By plugging SVM's loss function into the adaptation framework, we reach a specific adaptation algorithm named *Adaptive Support Machines* or *aSVM*.

4.3.1 Model Formulation

In aSVM, the delta function takes a linear form either in the original feature space as $\Delta f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w} \in \mathbb{R}^{d+1}$ are the parameters, or in a transformed feature space as $\Delta f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$, where $\phi(\cdot)$ is the feature map projecting the original feature $\mathbf{x}$ into the transformed space. In the latter case, $f^a(\mathbf{x})$ is in fact a non-linear function in the original feature space.

We adopt the hinge loss function of SVM which is expressed as $L(y, f(\mathbf{x})) = (1 - yf(\mathbf{x}))_+ = \max(1 - yf(\mathbf{x}), 0)$. Moreover, we set the regularizer to $\|\mathbf{w}\|^2$ by using a trivial regularization function $\Omega(x) = x$. The objective function of aSVM is given by plugging this loss function and regularizer into the adaptation framework in Eq.(2):

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(1 - y_i f(\mathbf{x}_i))_+ \tag{3}$$

This is equivalent to the following function:

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{s.t. } \ \xi_i \geq 0, \quad y_i f^a(\mathbf{x}_i) + y_i \mathbf{w}^T\phi(\mathbf{x}_i) \geq 1 - \xi_i, \ \ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}^t\end{aligned} \tag{4}$$

While this objective function is very similar to that of SVM (Eq.(12.8) in [37]), there is a fundamental difference: here $\mathbf{w}$ denotes the parameters of $\Delta f(\mathbf{x})$ instead of $f(\mathbf{x})$. In fact, we will show that $\|\mathbf{w}\|^2 = \|\Delta f\|^2_{\mathcal{H}} = \|f - f^a\|^2_{\mathcal{H}}$, which shows the regularizer is the distance between the auxiliary and target classifier in the function space, instead of "margin" in the case of SVM. Since $\sum_i \xi_i$ measures the classification error of the target classifier $f(\mathbf{x})$, the objective function in Eq.(5) seeks a classification boundary (hyperplane) that is close to the boundary of the auxiliary classifier, and is meanwhile able to correctly classify the labeled examples in $\mathcal{D}^t$. The cost factor C in aSVM balances the contribution between the auxiliary classifier (through the regularizer) and the training examples. Larger C indicates smaller influence of the auxiliary classifier, and vice versa. In practice, C should be decided based on the utility of auxiliary classifiers.

By integrating the constraints in Eq.(5) using Lagrange multipliers, we can rewrite the objective function as the following (primal) Lagrangian function:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\alpha_i(y_i f^a(\mathbf{x}_i) + y_i\mathbf{w}^T\phi(\mathbf{x}_i) - (1 - \xi_i)) \tag{5}$$

where $\alpha_i \geq 0, \mu_i \geq 0$ are Lagrange multipliers. We minimize L_P by setting its derivative with respect to $\mathbf{w}$ and ξ to zero, which results in:

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \phi(\mathbf{x}_i), \qquad \alpha_i = C - \mu_i, \ \ \forall i \tag{6}$$

From the above, it is easy to show that $\Delta f(\cdot) = \sum_{i=1}^{N}\alpha_i y_i K(\cdot, \mathbf{x}_i)$ as a function in the RKHS. Given the definition of inner product in RKHS, we can prove the regularizer $\|\mathbf{w}\|^2$ indeed measure the distance between the target classifier $f(\mathbf{x})$ and the auxiliary classifier $f^a(\mathbf{x})$ in RKHS.

$$\|f - f^a\|^2 = \|\Delta f\|^2 = \langle \Delta f, \Delta f\rangle \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{w}\|^2 \tag{7}$$

In addition to Eq.(6), the Karush-Kuhn-Tucker (KKT) conditions, which the optimal solution of Eq.(5) must satisfy, also include:

$$\begin{aligned}
\alpha_i\{y_i f^a(\mathbf{x}_i) + y_i \mathbf{w}^T \mathbf{x}_i - (1-\xi_i)\} &= 0 \\
\alpha_i &\geq 0 \\
y_i f^a(\mathbf{x}_i) + y_i \mathbf{w}^T \mathbf{x}_i - (1-\xi_i) &\geq 0 \\
\mu_i \xi_i &= 0 \\
\mu_i &\geq 0 \\
\xi_i &\geq 0
\end{aligned} \tag{8}$$

Substituting Eq.(6) into Eq.(5), we get the Lagrange dual objective function:

$$L_D = \sum_{i=1}^{N}(1-\lambda_i)\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

where $\lambda_i = y_i f^a(\mathbf{x}_i)$. The model parameters $\alpha\{\alpha_i\}_{i=1}^N$ can be estimated by maximizing L_D under the constraint $0 \leq \alpha_i \leq C, \forall i$. This would give a solution equivalent to that obtained by minimizing the primal function L_P. Maximizing L_D over α is a quadratic programming (QP) problem solved using the algorithm in Section 4.3.3. Given the solutions $\hat{\alpha}$, the target classifier is written as:

$$f(\mathbf{x}) \quad = \quad f^a(\mathbf{x}) + \sum_{i=1}^{N}\hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i) \tag{10}$$

where $(\mathbf{x}_i, y_i) \in \mathcal{D}^t$. The target classifier $f(\mathbf{x})$ can be seen as the auxiliary classifier $f^a(\mathbf{x})$ augmented with support vectors from the labeled examples of the target data.

4.3.2 Discussion

In this section, we discuss several key issues of aSVM in order to gain deeper insights of its properties and its connections/differences with other methods.

On support vectors. Support vectors of SVM are training examples that are on the classification boundary or on the wrong side of the boundary. Support vectors of aSVM have a different interpretation. We start by comparing the dual objective function of SVM (Eq.(12.13) in [37]) and aSVM in Eq.(9). The only difference is that the latter has $\{\lambda_i\}_{i=1}^N$ in the first term, where $\lambda_i = y_i f^a(\mathbf{x}_i)$. It is interesting to see how λ affect the estimation of α. In Eq.(9), if $\lambda_i = y_i f^a(\mathbf{x}_i) < 0$, which means the auxiliary classifier f^a misclassifies $\mathbf{x}_i$, a larger α_i is desired in order to maximize L_D, and vice versa. This is intuitive because the target classifier f is adapted from f^a with the support vectors $\mathbf{x}_i \in \mathcal{D}^t$, and α_i can be seen as the weight of each support vector. If the auxiliary classifier f^a misclassifies $\mathbf{x}_i$, which means the boundary of f^a around $\mathbf{x}_i$ is wrong, then the boundary of the target classifier f around $\mathbf{x}_i$ needs to made different from f^a in order to correctly classify $\mathbf{x}_i$. This is realized by adding $\mathbf{x}_i$ as a support vector with a large weight α_i. On the contrary, if the auxiliary classifier correctly classifies $\mathbf{x}_i$, $f(\mathbf{x}_i)$ does not need to be different from $f^a(\mathbf{x}_i)$, so the weight α_i can be small or even zero. This shows that the support vectors in aSVM are used to *correct the misclassifications* of the auxiliary classifier.

On training cost. A key benefit of function-level adaptation is high efficiency as the result of avoiding training over auxiliary data. We show why it is the case in aSVM. It is clear form from Eq.(9) that the number of parameters $\{\alpha_i\}_{i=1}^N$ in aSVM is equal to the number of target examples N, and not related to the number of auxiliary data N^a. It has the same number of parameters as a standard SVM model trained from $\mathcal{D}^t$. Thus, adapting $f^a(\mathbf{x})$ to $f(\mathbf{x})$ using aSVM is no more expensive than training a SVM model entirely from $\mathcal{D}^t$, except the cost associated with computing $\{\lambda_i\}_{i=1}^N$. Since $\lambda_i = y_i f^a(\mathbf{x}_i)$ remains as a constant throughout the optimization process (see Section 4.3.3), this is one-time cost of evaluating $f^a(\mathbf{x})$ for N data instances in $\mathcal{D}^t$.

While the actual cost of $\{f^a(\mathbf{x}_i)\}_{i=1}^N$ depends on the complexity of the auxiliary classifier f^a, it is linear with N. In comparison, even the most efficient training methods for SVM, such as SVM-Light [43], SMO [59], LIBSVM [15], all have superlinear scaling factor with N. In [43], it has been shown that the time complexity of SVM is $O(N^k)$ where $k \approx 1.7$ for real-valued feature vectors. So the complexity of training an aSVM model and training a SVM model from $\mathcal{D}^t$ are both $O(N^k)$. This is considerably smaller than the $O((N+N^a)^k)$ complexity of training a SVM model over all training examples $\mathcal{D}^a \bigcup \mathcal{D}^t$, because the size of auxiliary data is typically much larger than that of labeled target data, i.e., $N^a >> N$. The experiments to be presented support this analysis.

On cost factor C. In aSVM, C balances the classification error and the deviation from the auxiliary classifier f^a, with large C emphasizing small classification error and small C emphasizing closeness to f^a. Intuitively, one should use smaller C for "better" auxiliary classifiers that work well on the target data, and vice versa. That being said, in practice the absolute value of C is also influenced by the range of $f^a(\cdot)$. Since $0 \leq \alpha_i \leq C$, the range of delta function $\Delta f(\mathbf{x}) = \sum_{i=1}^N \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i)$ is constrained by C. If the range of $f^a(\cdot)$ is large, say, $[-100, +100]$, C needs to be large enough so that $\Delta f(\mathbf{x})$ is not overwhelmed by $f^a(\mathbf{x})$ when they add up to form the target classifier $f(\mathbf{x})$ (see Eq.(10)). Another possibility is to normalize $f^a(\cdot)$ to a fixed range such as $[0, 1]$. We will explore such engineering issues in the experiments.

Comparison with ensemble and increment learning. The decision function $f(x)$ defined by Eq.(10) has an additive form, which is similar to an ensemble classifier that combines the auxiliary classifier $f^a(x)$ and the delta function $\Delta f(x)$. However, it is different from a genuine ensemble classifier where the component classifiers are trained *independently* from different datasets ($\mathcal{D}^a$ and $\mathcal{D}^t$ in our case). It is clear from Eq.(9) that α as parameters of $\Delta f(\cdot)$ is estimated under the influence of $f^a(\cdot)$, so its value would be different if it is estimated *exclusively* from $\mathcal{D}^t$. Also, aSVM should not be mixed up with some incremental learning algorithms for SVM, such as those proposed by Syed et al. [78] and by Cauwenberghs and Poggio [14]. Although these methods add support vectors from new data into the existing SVM model, they also modify the original support vectors in the existing model. In comparison, aSVM does not change the original support vectors in the auxiliary classifier.

4.3.3 Learning Algorithm of aSVM

The parameters α of aSVM are estimated by maximizing the dual objective function defined in Eq.(9). This is a quadratic programming (QP) problem, where the number of variables $\{\alpha_i\}_{i=1}^N$ is equal to the number of labeled examples in D^t. The sequential minimal optimization (SMO) algorithm proposed by Platt [59] can efficiently solve a large QP problem by decomposing it into a series of QP subproblems and optimizing them iteratively. We modify the original SMO algorithm to solve aSVM efficiently.

Before the technical details, it is worthwhile to note an important difference between the SMO algorithm for aSVM and that for standard SVM. For SVM, the minimum sub-problem tackled in each iteration of SMO optimizes *two* variables. One cannot optimize a single variable at a time because of the linear constraint $\sum_i \alpha_i y_i = 0$ derived from $\frac{\partial L_P}{\partial b} = 0$, where b is the intercept in a decision function $f(x) = \mathbf{w}^T \phi(x) + b$. In contrast, such constraint does not exist in aSVM and therefore its SMO algorithm optimizes *only one* variable in each iteration. On the surface, as the constraint is derived from $\frac{\partial L_P}{\partial b} = 0$, its absence is because our decision function $f(x) = \mathbf{w}^T \phi(x)$ does not explicitly represent the intercept b but implicitly includes it as a component of $\mathbf{w}$. This notational difference is not critical. The real reason is that in aSVM, the intercept is implicitly involved in the regularizer $\|\mathbf{w}\|^2$, while in SVM the intercept is *not* part of the regularizer (since it does not affect margin). Even if we explicitly include b in $f(x)$, one cannot derive this linear constraint as long as b is included as part of regularizer, say, in the form of $\|b\|^2$.

The parameter α is the optimal solution to the QP problem in Eq.(9) *if and only if* the KKT conditions in Eq.(8) are fulfilled. We decompose the optimality condition in Eq.(8) according to the value of α_i:

$$\begin{aligned} \alpha_i = 0 \quad &\Rightarrow \quad \mu_i = C, \xi_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1 \\ 0 < \alpha_i < C \quad &\Rightarrow \quad \mu_i > 0, \xi_i = 0 \Rightarrow y_i f(\mathbf{x}_i) = 1 \\ \alpha_i = C \quad &\Rightarrow \quad \mu_i = 0, \xi_i \geq 0 \Rightarrow y_i f(\mathbf{x}_i) \leq 1 \end{aligned} \tag{11}$$

If the above optimal conditions are satisfied for every i, the QP problem is solved, otherwise it is not. Eq.(11) provides a method to check the optimality condition of the problem, and also to find variables α_i that violate such condition and need to be optimized. Our SMO algorithm chooses one variable to optimize in each iteration. While there are many ways to select working variables, we use an intuitive heuristic of selecting the variable α_{i^*} that violates the optimality condition the most:

$$\begin{aligned} i^* = &\operatorname*{argmax}_{i \in \{i_{up}, i_{low}\}} |y_i f(\mathbf{x}_i) - 1| \\ \text{where} \quad i_{up} = &\operatorname*{argmin}_{i \in \{t | \alpha_t < C\}} y_i f(\mathbf{x}_i), i_{low} = \operatorname*{argmax}_{i \in \{t | \alpha_t > 0\}} y_i f(\mathbf{x}_i) \end{aligned} \tag{12}$$

Without loss of generality, suppose α_1 is the working variable to optimize. We update it by setting the derivative of the dual objective function L_D against α_1 to zero:

$$\frac{\partial L_D}{\partial \alpha_1} = 1 - y_1 f^{old}(x_1) - y_1(\alpha_1^{new} - \alpha_1^{old}) K(x_1, x_1) = 0$$

where $f^{old}(x)$ is the target classifier Eq.(10) evaluated using the existing value of α. This leads to an analytical solution of α_1:

$$\alpha_1^{new} = \alpha_1^{old} + \frac{1 - y_1 f^{old}(x_1)}{K(x_1, x_1)}$$

Due to $0 \leq \alpha_i \leq C$, the constrained optimal of α_1 is given by clipping the unconstrained optimal using the following bounds:

$$\alpha_1^{new,clipped} = \begin{cases} C, & \text{if } \alpha_1^{new} \geq C; \\ \alpha_1^{new}, & \text{if } 0 < \alpha_1^{new} < C; \\ 0, & \text{if } \alpha_1^{new} \leq 0 \end{cases} \tag{13}$$

To summarize the SMO learning algorithm, we start with certain initializations of α, and iteratively choose working variables using Eq.(12) and optimize them one at a time using Eq.(13). This process continues until the optimality condition in Eq.(11) is satisfied up to a certain accuracy.

4.4 Adaptive Kernel Logistic Regression (aKLR)

In this section, we derive *Adaptive Kernel Logistic Regression* or *aKLR* from our adaptation framework by adopting the logistic loss function. We will present an alternative, probabilistic interpretation of aKLR, where it is derived from the maximum-a-posterior (MAP) estimation of a kernel logistic regression (KLR) model with a Gaussian Process prior.

4.4.1 Model Formulation

To derive aKLR from the adaptation framework in Eq.(2), we adopt the logistic loss function $L(y, f(\mathbf{x}) = \log(1 + \exp(-yf(\mathbf{x})))$ used by (kernel) logistic regression, and set the delta function as $\Delta f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$ and the regularizer as $\|\mathbf{w}\|^2$. This leads to the following objective function:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \log(1 + \exp(-y_i f(\mathbf{x}_i))) \tag{14}$$

By defining $\xi_i = -y_i f(\mathbf{x}_i)$, we can rewrite it as the (primal) Lagrangian form:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \log(1 + e^{\xi_i}) - \sum_{i=1}^{N} \alpha_i(\xi_i + y_i f^a(\mathbf{x}_i) + y_i \mathbf{w}^T \mathbf{x}_i) \tag{15}$$

where $\{\alpha_i\}_{i=1}^N$ are Lagrange multipliers. To minimize L_P, we set its derivative against $\mathbf{w}$ and ξ_i to zero, which gives $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i)$ and $\xi_i = \log \frac{\alpha_i}{C-\alpha_i}$. Plugging them into the primal form in Eq.(15), we have the dual Lagrangian function:

$$\begin{aligned} L_D \;=\; & -\sum_{i=1}^{N} \alpha_i y_i f^a(\mathbf{x}_i) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & -\sum_{i=1}^{N}(C - \alpha_i)\log(C - \alpha_i) - \sum_{i=1}^{N} \alpha_i \log \alpha_i \end{aligned} \tag{16}$$

The parameters $\{\alpha_i\}_{i=1}^N$ are estimated by maximizing the above dual form using the learning algorithm to be described in Section 4.4.3. The target classifier has exactly the same form as that of aSVM, i.e., $f(\mathbf{x}) = f^a(\mathbf{x}) + \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$.

4.4.2 A Probabilistic Interpretation

We show that aKLR can be also derived from a probabilistic perspective where the auxiliary classifier is treated as "prior model" of the target classifier. This alternative derivation is based on *maximum-a-posterior* (MAP) estimation. Following the representation of (kernel) logistic regression [37], we define the conditional probability as $p(y|\mathbf{x}) = 1/(1+\exp(-yf(\mathbf{x})))$. Thus, the log-likelihood of the training examples is represented as:

$$l(\mathcal{D}^t; f) = \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i) = -\sum_{i=1}^{N} \log(1+e^{\xi_i}) \tag{17}$$

where $\xi_i = -y_i f(\mathbf{x}_i)$. This log-likelihood corresponds to the second term of aKLR's objective function in Eq.(14), which represents empirical loss.

From a Bayesian point of view, the regularizer $\|\mathbf{w}\|^2$ is the outcome of a Guassian prior distribution on parameters, i.e., $\mathbf{w} \sim \mathcal{N}(0, C\mathbf{I})$, where $\mathbf{I}$ is an identity matrix. The distribution $p(\mathbf{w})$ specifies the properties of the considered function space of $f(\mathbf{x})$ in terms of the *mean function*:

$$E(f(\mathbf{x})) = f^a(\mathbf{x}) + E(\mathbf{w}^T)\phi(\mathbf{x}) = f^a(\mathbf{x})$$

and *covariance function*:

$$Cov(f(\mathbf{x}), f(\mathbf{x}')) = E(\mathbf{w}^T\phi(\mathbf{x}) \cdot \mathbf{w}^T\phi(\mathbf{x}')) = Var(\mathbf{w})K(\mathbf{x},\mathbf{x}') = CK(\mathbf{x},\mathbf{x}')$$

This means the target classifier $f(\mathbf{x})$ follows a prior *Gaussian process* (GP) [64] with mean function $f^a(x)$ and its covariance function $CK(\mathbf{x},\mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}(f^a(\mathbf{x}), CK(\mathbf{x},\mathbf{x}'))$$

GP is a distribution in the function space. This GP prior in particular favors target classifiers $f(\mathbf{x})$ that are close to the auxiliary classifier $f^a(\mathbf{x})$ in the function space. According to the definition of GP [64], a finite sample of $f(\mathbf{x})$ as $\mathbf{f} = \{f(\mathbf{x}_1), .., f(\mathbf{x}_N)\}$, where $\mathbf{x}_i \in \mathcal{D}^t$, follow a joint Gaussian distribution with mean as $\mathbf{f}^a = \{f^a(\mathbf{x}_1), .., \mathbf{f}^a(\mathbf{x}_N)\}$ and covariance matrix as $C\mathbf{K}$ where $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{N\times N}$, i.e., $\mathbf{f} \sim \mathcal{N}(\mathbf{f}^a, C\mathbf{K})$.

To learn the target classifier $f(\mathbf{x})$, we resort to the *maximum-a-posterior* (MAP) estimation instead of a fully Bayesian approach. If we treat $\mathbf{f}$ as the model parameters, the logarithm of the posterior distribution is specified by the log-likelihood of data and the prior $p(\mathbf{f})$:

$$\begin{aligned} \log p(\mathbf{f}|D) &\propto l(D;\mathbf{f}) + \log p(\mathbf{f}) \\ &= -\sum_{i=1}^{N} \log(1+e^{\xi_i}) - \frac{1}{2C}(\mathbf{f}-\mathbf{f}^a)^T\mathbf{K}^{-1}(\mathbf{f}-\mathbf{f}^a) + const \\ &= -\sum_{i=1}^{N} \log(1+e^{\xi_i}) - \frac{1}{2C}\|\mathbf{w}\|^2 + const \end{aligned} \tag{18}$$

which is to be maximized subject to $\xi_i = -y_i f(\mathbf{x}_i)$. This maximization problem is identical to the minimization problem of aKLR in Eq.(14). This shows that the aKLR model can be equivalently derived from a fully probabilistic perspective.

This probabilistic interpretation does not extend to other methods under our adaptation framework, such as aSVM, because their loss functions cannot be interpreted as conditional probabilities. Nevertheless, it sheds light on the role of auxiliary classifier in this framework, which is a prior in the hypothesis (function) space for the target classifier $f(\mathbf{x})$. The regularizer $\|\Delta f\|_{\mathcal{H}}$ penalizes any hypothesis $f(\mathbf{x})$ that deviates from this prior.

4.4.3 Learning Algorithm for aKLR

Similar to aSVM, the parameters of aKLR are also estimated by minimizing its dual Lagrangian form L_D in Eq.(16). This is a quadratic programming (QP) problem solved using another variation of SMO algorithm we proposed. It is inspired by the SMO algorithm for kernel logistic regression proposed by Keerthi et al. [44].

For simplicity, we define $F_i = -y_i f(\mathbf{x}_i) + \log(C - \alpha_i) - \log \alpha_i$. To maximize L_D defined in Eq.(16), we can set its derivative against every α_i to zero:

$$\frac{\partial L_D}{\partial \alpha_i} = -y_i f(\mathbf{x}_i) + \log(C - \alpha_i) - \log \alpha_i = F_i \triangleq 0 \tag{19}$$

This provides a method for checking the optimality condition (i.e., $F_i = 0, \forall i$) and for finding variables that violate the condition. Following the same reason discussed in Section 4.3.3, only one working variable is optimized in each iteration of this SMO algorithm. We select the one that violates the optimality condition the most as our working variable, i.e., $i* = \text{argmax}_i |F_i|$.

Without loss of generality, let α_1 be the current working variable. Unlike in aSVM, here we cannot derive an analytical solution of α_1 from Eq.(19), so we resort to the Newton-Raphson (NR) method to optimize α_1 iteratively. Suppose $\alpha_1^{new} = \alpha_1^{old} + t$. In the NR method, t is iteratively updated using the following equation:

$$t^{l+1} = t^l - \frac{\partial L_D}{\partial t}\left(\frac{\partial^2 L_D}{\partial t^2}\right)^{-1} \tag{20}$$

where

$$\begin{aligned}
\frac{\partial L_D}{\partial t} &= \frac{\partial L_D}{\partial \alpha_1}\frac{\partial \alpha_1}{\partial t} = -y_1 f^{old}(\mathbf{x}_1) - tK(\mathbf{x}_1, \mathbf{x}_1) + \log(C - \alpha_1^{old} - t) - \log(\alpha_1^{old} + t) \\
\frac{\partial^2 L_D}{\partial t^2} &= -K(\mathbf{x}_1, \mathbf{x}_1) - \frac{1}{C - \alpha_1^{old} - t} - \frac{1}{\alpha_1^{old} + t}
\end{aligned}$$

Starting from $t^0 = 0$, we repeatedly invoke Eq.(20) to update t until a certain accuracy is reached or the constraint $0 < \alpha_1 < C$ becomes tight. The resulting t is used to update α_1. We iteratively optimize selected working variables until the optimality condition is satisfied, i.e., $F_i = 0, \forall i$.

4.5 Preliminary Experiments

We have applied aSVM and aKLR to the problem of adapting classifiers for semantic video concepts across different news video corpora and demonstrated their effectiveness in comparison with several reference methods in accuracy and efficiency.

4.5.1 Approaches for Comparison

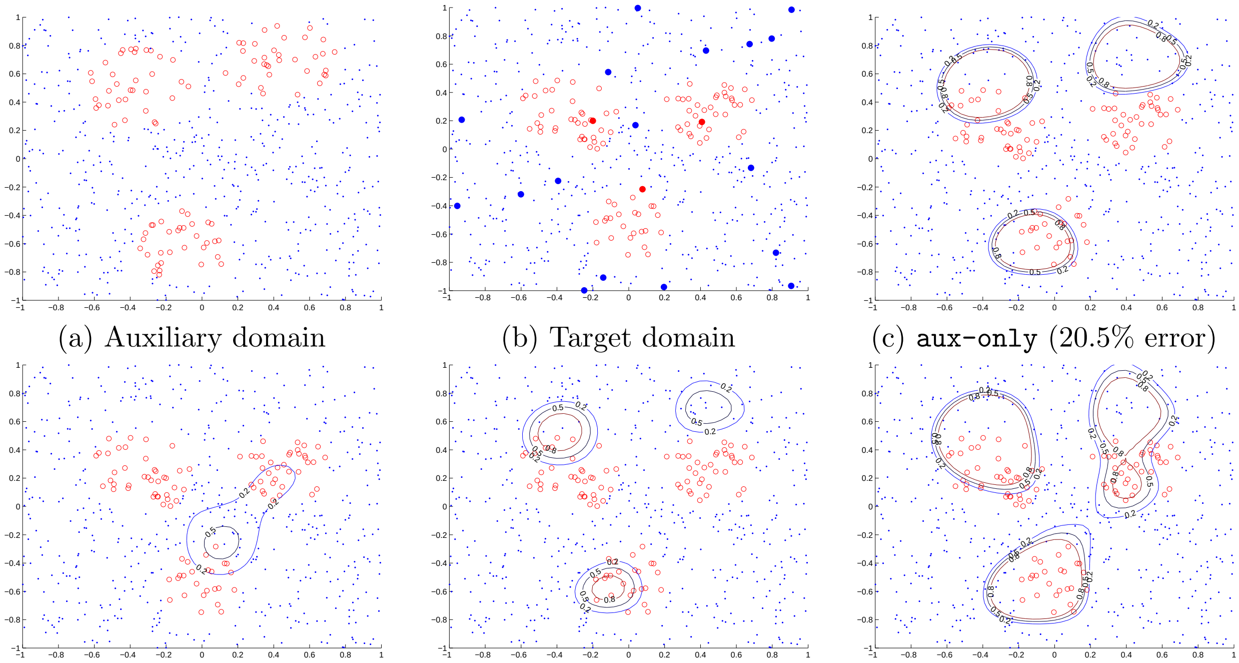
We discuss four alternative approaches for classifying the data in the target domain. The first two are baseline approaches, while the last two methods can be viewed as adaptation approaches because they exploits the knowledge in both the auxiliary and target domain.

- **Aux-only approach:** As suggest by its name, this approach directly applies the classifier $f^a(\mathbf{x})$ (SVM or KLR) trained from an auxiliary domain on the target domain. The cost factor used in training is C^a.
- **Target-only approach:** Opposite to the first approach, this approach ignores the auxiliary domain and builds SVM or KLR classifier $f^t(\mathbf{x})$ entirely from the labeled examples $\mathcal{D}^t$ in the target domain. We use C to denote the cost factor used in training.
- **Aggregation approach:** This computationally intensive approach learns a single SVM or KLR classifier using *all* the labeled examples aggregated from the auxiliary domain and the target domain, i.e., $\mathcal{D}^a \bigcup \mathcal{D}^t$. The examples from the two domains are weighted according to their relative importance to the task by using different cost factors in SVM or in KLR algorithm. We modify the implementation for SVM and KLR to support different cost factors of examples. In the case of SVM, the decision function is $f^{aggr}(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$ with $\mathbf{w}$ estimated from the following objective function:

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i + C^a\sum_{i=1}^{N^a}\xi_i^a \qquad (21) \\ \text{s.t.} \quad & \xi_i \geq 0, \quad y_i\mathbf{w}^T\phi(\mathbf{x}_i) \geq 1 - \xi_i, \ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}^t \\ & \xi_i^a \geq 0, \quad y_i^a\mathbf{w}^T\phi(\mathbf{x}_i^a) \geq 1 - \xi_i^a, \ \forall(\mathbf{x}_i^a, y_i^a) \in \mathcal{D}^a \end{aligned}$$

  where C and C^a are the cost factors for examples in the target domain and in the auxiliary domain, respectively. Intuitively, we we have $C > C^a$ because the goal is to classify the target domain well. The aggregation approach based on KLR has a similar form. This approach represents the data-level transfer learning approaches used in existing works [24, 46, 50, 88]. It is fundamentally different from our adaptation approach in that it involves the auxiliary data in the training process while our approach directly manipulate the auxiliary classifiers.
- **Ensemble approach:** While the aggregation approach combines the training examples, the ensemble approach combines the output of two classifiers, one trained from the auxiliary domain and the other trained from the target domain. Its result is equal to a weighted sum of output of the target-only method and the aux-only method:

$$f^{ens}(\mathbf{x}) = Cf^t(\mathbf{x}) + C^a f^a(\mathbf{x}) \qquad (22)$$



(a) Auxiliary domain (b) Target domain (c) `aux-only` (20.5% error)

(d) `target-only` (16.7% error) (e) `aggregation` (16.8% error) (f) `adaptation` (15.0% error)

Figure 5: (a) Distribution of D_A; (b) Distribution of D_T; (c)-(f) Decision boundaries of the classifiers trained using `aux-only`, `target-only`, `aggregation` and `adaptation` method and applied to D_T. The number in the parentheses show the error rate of each classifier.

where C and C^a are used as the weights of the two classifiers. This has been the approach used in [47, 86]. Note that although our adapted classifier takes a similar ensemble form $f(\mathbf{x}) = f^a(\mathbf{x}) + \Delta f(\mathbf{x})$, it is different from this ensemble approach since $\Delta f(\mathbf{x})$ is learned with the influence of $f^a(\mathbf{x})$. In contrast, in this ensemble approach the two base classifiers $f^a(\mathbf{x})$ and $f^t(\mathbf{x})$ are learned independently.

- **Adaptation approach:** We use the proposed aSVM or aKLR algorithm to adapt the classifier $f^a(\mathbf{x})$ trained from the auxiliary domain to a new classifier $f(\mathbf{x})$ based on the labeled target examples $\mathcal{D}^t$. The cost factor in aSVM or aKLR is set to C.

Cares are taken to ensure these approaches and our adaptation approach are comparable. In experiments, they are always trained using the same learning algorithms, either SVM (and aSVM) or KLR (and aKLR), and the same kernel function, which we use the empirically successful RBF kernel $K(x_i, x_j) = e^{-\rho\|x_i - x_j\|^2}$. We also ensure that the cost factor C^a for the auxiliary domain as well as C^a for the target domain are the same across different approaches. For simplicity, we set $C^a = 1$ in all the experiments, and set C to different values to study its impact on the classification performance.

4.5.2 Performance: Synthetic Data

To illustrate our model, we generate two synthetic data sets D_A and D_T from different distributions in a 2-d feature space to represent the data in the auxiliary and target domain, respectively. Each set has 100 positive and 500 negative data. The positive data in each set are generated from a Gaussian mixture model with 3 Gaussian components, and the negative data are sampled uniformly outside the area of the positive data. For D_A, the 3 Gaussian components are centered at $(-0.4, 0.5)$, $(0.5, 0.7)$, and $(-0.1, -0.6)$, while for D_T their means shift to $(-0.4, 0.3)$, $(0.5, 0.3)$, and $(0, -0.65)$.

Figure 5 (a) and (b) shows the distribution of D_A and D_T, where small circles denote positive instances and dots denote negative instances. We assume *all* the instances in D_A are labeled, while only 20 instances are labeled in D_T, including 3 positive instances and 17 negative instances, shown as large dots in Figure 5 (b). In the task of classifying D_T in the target domain, we compare our adaptation approach based on aSVM with the `target-only`, `aux-only`, `aggregation`, and `ensemble` approach mentioned above.

We plot the decision boundary of the four classifiers on D_T in Figure 5 (c) to (f). The error rate of each classifier is shown below each figure. Not surprisingly, the boundary of `aux-only` classifier is biased towards the distribution of D_A, and it is unable to discriminate the positive data in D_T whose positions have shifted. The `target-only` approach is unbiased, but has a large variance due to the limited training data. `aux-only` and `target-only` have the two worst error rates, showing that using evidence from a single domain is inadequate. Although trained from examples in both domains, the `aggregation` method is still biased perhaps because the examples from D_T are outnumbered by those from D_A. The `adaptation` method achieves the lowest error rate, and its decision boundary captures the distribution of the positive data in D_T more precisely than the other approaches. We attribute its superior performance to a good bias-variance tradeoff: the model has a low bias by training from only the examples of D_T, and meanwhile achieves a low variance through a regularizer penalizing its difference from a prior model.

4.5.3 Performance: Cross-channel Video Concept Detection

The experiment on cross-domain video concept detection is conducted based on the TRECVID 2005 collection described in Section 3. This collection contains news video from 6 different channels (broadcasters) and manual annotations of 39 semantic concepts. We choose one channel as the *target channel* and another as the *auxiliary channel*. We assume *all* the video shots in the auxiliary channel are labeled w.r.t the 39 concepts, based on which a concept classifier has been trained from each concept. The video shots in the target channel are partitioned into a *development set* and a *test set*. In temporal order, the first 40% of the shots belong to the development set, and the remaining 60% belong to the test set. We label a certain number of shots *randomly* selected from the development set, and treat them as the training examples. The shots in the test set are unlabeled and used for evaluation purpose.

The goal is to adapt the classifier of a given concept trained from the auxiliary channel to a classifier for the target channel based on the training examples in its development set. The adapted classifier is evaluated on the test set using the average precision (AP) metric. In the experiment, we iterative over all the 39 concepts except the "Weather" concept since its number of relevant shots is too small. For each concept, we repeat the experiment using 4 sets of random samples in order to reduce the variance caused by random sampling. The overall performance is measured by mean average precision (MAP) averaged from 38 concepts and 4 random iterations for each concept.

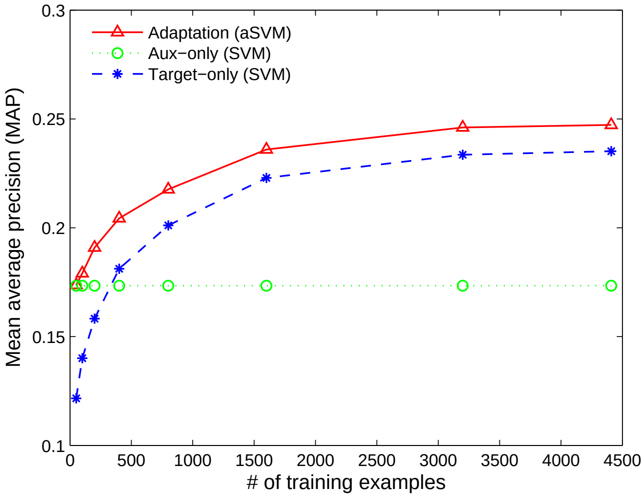


Figure 6: Comparison of our adaptation approach with two baseline approaches in terms of average performance of detecting 38 concepts in CNN channel using NBC as auxiliary channel.

In this experiment we choose CNN as the target channel and NBC as the auxiliary channel.

We compare five approaches including the `adaptation` approach based on aSVM, and the `target-only`, `aux-only`, `aggregation`, and `ensemble` approach, where the last four approaches are based on SVM. All the approaches use the same cost factor $C = 1$ and $C^a = 1$. Moreover, they all use the Radius Basis Function (RBF) $K(x_i, x_j) = e^{-\rho\|x_i - x_j\|^2}$ as the kernel function with $\rho = 0.1$.

Comparison with Baseline Methods: Figure 6 compares our `adaptation` approach based on aSVM and the `target-only` and `aux-only` approach in terms of MAP across 38 concepts. We vary the number of training examples in the target channel from 50 to 4410, which is the size of the entire development set, each time doubling the number of examples. Because the average frequency of a concept is only 7%, which means only 7 out of 100 labeled shots are relevant to a concept, the positive examples actually labeled are relatively few and far below the amount needed to train reliable classifiers. So the numbers of labeled examples in this experiment are limited compared with existing works [1, 38, 74].

We see that on average the `adaptation` approach outperforms the two baseline approaches by a substantial margin. The performance gain over `target-only`, which uses a new classifier trained without the knowledge of the auxiliary channel, is significant especially when the training examples are relatively scarce. The gain diminishes as the number of the training examples increases, but does not fade out even when all the data in the development set are used for training. This shows adaptation is beneficial even when there is a relatively large amount of training data. From another perspective, `adaptation` needs much less training data to reach the same performance. For example, `target-only` needs 400 labeled examples to achieve the same MAP of `adaptation` using only 50 examples, or 1600 examples to reach the MAP of `adaptation` at 400 examples. This clearly shows the benefit of leveraging the knowledge of

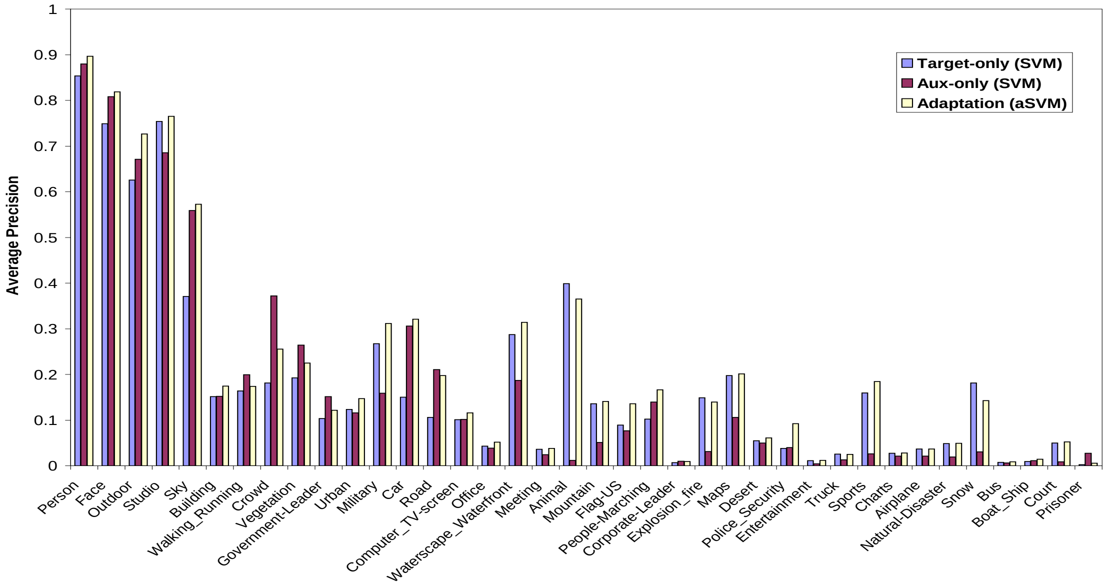


Figure 7: Performance of detecting each concept in CNN using NBC as auxiliary channel.

Concept frequency		High(>10%)	Medium	Low(<1%)	Overall
# of concepts		6	20	12	38
Best Performer	`target-only`	0	2	2	4 (10.5%)
	`aux-only`	0	6	1	7 (18.4%)
	`adaptation`	6	12	9	27 (71.1%)
Worst Performer	`target-only`	5	9	3	17 (44.7%)
	`aux-only`	1	11	9	21 (52.6%)
	`adaptation`	0	0	0	0 (0%)

Table 1: The distribution of concepts by the performance of three approaches.

auxiliary channel (classifiers) in terms of improving performance or reducing labeling effort. In contrast, the performance gain of the adapted classifier (`adaptation`) over the auxiliary classifier (`aux-only`) enlarges as more training examples become available. This shows that aSVM can effectively exploit the knowledge in training examples to improve existing classifiers.

We also compare the three approaches on a per concept basis in order to examine the consistency of the performance improvement from adaptation. Figure 7 compares their performance in terms of AP on the 38 concepts in descending order of frequency, when there are 400 training examples available. In addition, Table 1 shows the number of concepts on which each approach has the best or worst performance in each frequency range. We see that the `adaptation` approach is the best performer on majority of the concepts, and a close runner-up in the remaining concepts. It never has worst performance in any concept. Since we do not know *a-priori* whether `target-only` or `aux-only` performs better, this result means that `adaptation` is able to (implicitly) pick the best of the two and improve on it. So our approach is relatively consistent

# of samples	Methods				
	`target-only`	`aux-only`	`ensemble`	`aggregation`	`adaptation`
50	0.122	0.173	0.165	0.171	0.174
100	0.140	0.173	0.171	0.183	0.179
200	0.158	0.173	0.182	0.191	0.191
400	0.181	0.173	0.194	0.205	0.204

Table 2: Performance comparison of different methods.

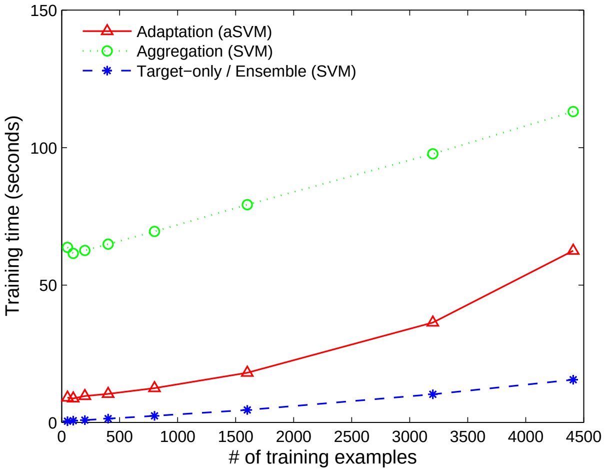


Figure 8: Comparison of average training time of 38 concepts

across different concepts, although there is still room for improvement.

A closer examination shows that the `adaptation` approach exploits different types of knowledge depending on the frequency of a concept. For frequent concepts, the auxiliary classifiers generalize well and perform better than new classifiers trained from limited labeled examples. In this case, our approach mainly relies on the auxiliary classifiers and further improve their performance through adaptation. For rare concepts, the auxiliary classifiers perform poorly with the worst performance on 9 out 12 concepts, and the training examples become more valuable. In this case, the `adaptation` approach can fully utilize the training examples without affected by the poor auxiliary classifiers. This demonstrates that our approach is able to take advantage of "the best of two worlds" in any situations.

Comparison with competing approaches: We also compare our method to two competing methods that exploit both the auxiliary and target channel: the `aggregation` method, which trains concept classifiers from labeled data aggregated from the two channels,and the `ensemble` approach, which combines the outputs of classifiers trained separately from the two channels. Table 3 shows the performance of them and the two baseline approaches in terms of MAP of detecting 38 concepts. Clearly, the three methods leveraging the knowledge from both channels significantly outperforms those using knowledge from a single channel. Among the top three, `adaptation` and `aggregation` have comparable performance, which is better than the performance of `ensemble`.

# of samples	**Auxiliary channel**				
	NBC	MSNBC	LBC	CCTV	NTDTV
0 (no adaptation)	0.173	0.197	0.151	0.146	0.131
50	0.181	0.190	0.165	0.165	0.155
100	0.192	0.199	0.178	0.179	0.168
200	0.196	0.207	0.193	0.189	0.180
400	0.205	0.213	0.202	0.199	0.190

Table 3: The performance of aSVM using different auxiliary channels.

# of samples	**Cost factor C**			
	C=1	C=2	C=3	C=5
50	0.174	0.166	0.161	0.155
100	0.179	0.172	0.169	0.164
200	0.191	0.185	0.181	0.177
400	0.204	0.201	0.198	0.196

Table 4: The performance of aSVM with different cost factors.

Training cost is another important performance metric that our adaptation approach aims to improve. We compare the total training time of `adaptation`, `aggregation`, `ensemble`, and `target-only` in Figure 4. We assume auxiliary classifiers are trained "offline", and consequently, the cost of `ensemble` is equal to that of `target-only`. From Figure 4, we see that the training time of `adaptation` is considerably lower than `aggregation` especially at small sample size. This is due to the fact that the `aggregation` method involves a large amount of auxiliary data in training, while the `adaptation` method does not. Therefore, our aSVM-based adaptation method achieves a good balance between efficiency and classification performance.

Impact of domain relatedness: While we use a fixed auxiliary channel in the experiments, it is very interesting to investigate how the choice of auxiliary channel (classifier) affects performance. With CNN being the target channel, we vary our choice of auxiliary channel between the other five channels, and for each choice evaluate the aSVM-based adaptation approach using the setting described above. As shown in Table 3, the performance of adaptation is largely influenced by the choice of auxiliary channels, and is closely related to the performance of auxiliary classifiers (before adaptation) on the target channel. In general, better auxiliary classifiers lead to better adapted classifiers by providing a higher starting point for adaptation. When more training examples are available, the performance difference between auxiliary channels becomes less substantial.

Besides choosing auxiliary channels, we can adjust the contribution of an auxiliary classifier in aSVM through the cost factor C. Based on the discussion in Section 4.3, smaller C puts more emphasis on the auxiliary classifiers, and vice versa. From Table 4 we see that the choice of C also has a moderate impact on the performance of our adaptation approach. Although its impact is not as large as that of the choice of auxiliary channel, one needs to choose a proper cost factor in order to maximally utilize the chosen auxiliary channel. Currently, C can be only chosen through cross-validation which is computationally expensive.

Choosing auxiliary classifiers and choosing cost factor are two *orthogonal* aspects of an important issue: measuring the utility of an auxiliary classifier in adaptation. Our adaptation approach is limited as each auxiliary classifier is treated equally without considering its utility. While the experiment suggests to select auxiliary classifiers with better performance on the target data, measuring such performance is challenging given that the target domain is mostly unlabeled. We will address this issue using domain analysis approaches in Section 5.

5 Domain Analysis

The proposed adaptation framework and algorithms do not consider how much the auxiliary domain is related to the target domain. They provide a mechanism for doing the adaptation, without asking whether the adaptation should be done or how much emphasis should be given to the auxiliary classifiers. This is a limitation because adaptation is beneficial only when the auxiliary and target domain is *sufficiently related.* Classifiers adapted from irrelevant auxiliary domains can perform worse than classifiers trained only from the target domain. Another limitation is that only one auxiliary classifier is used in adaptation, while in practice more than one auxiliary domains may provide useful knowledge for problems in the target domain. For example, to build concept classifiers for CCTV video, one may find the classifiers for CNN, MSNBC, and NTDTV video helpful. Therefore, it is desired to support adaptation of multiple auxiliary classifiers, as well as to include weight for each classifier to reflect its (projected) contribution to the target domain. In this section, we extend our adaptation framework to accommodate multiple auxiliary classifiers, and explore several *domain analysis* approaches to measure the relatedness between auxiliary and target domains in order to weight the contribution of auxiliary classifiers accordingly.

5.1 Proposed Work: Adaptation with Domain Weighting

5.1.1 Extended Framework

We deal with multiple auxiliary classifiers and their weighting by constructing a weighted combination, or an *ensemble*, of the outputs of the auxiliary classifiers $\sum_k t_k f_k^a(\mathbf{x})$ and treat it as a single classifier to be adapted to the target classifier. Here, $\{f_k^a\}_{k=1}^M$ denote classifiers trained separately from M auxiliary domains, and $\mathbf{t} = \{t_k\}_{k=1}^M$ are their weights. The weights reflect the utility of each auxiliary classifier, which is in turn determined by the relatedness between each auxiliary domain and the target domain. In this case, the target classifier is represented as:

$$f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \Delta f(\mathbf{x}) \tag{23}$$

The weights $\{t_k\}_{k=1}^M$ can be set manually based on our knowledge about the relatedness of each auxiliary domain to the target domain. In this case, the ensemble $\sum_k t_k f_k^a(\mathbf{x})$ is *fixed* before the adaptation process, and we can use exactly the same approach in Section 4 to adapt it to the target classifier. However, manually defining the weights is not desirable because the human knowledge about domain relatedness can be inaccurate and often unavailable.

In this section, we extend our framework in Section 4 so that the weights $\{t_k\}_{k=1}^M$ of auxiliary classifiers can be learned *automatically* and *simultaneously* with the target classifier in one unified learning process. This is realized by adding another regularizer $\Psi(\|\mathbf{t}\|)$ to the regularized loss minimization framework:

$$\min_{\Delta f, \mathbf{t}} \sum_i L(y_i, f(\mathbf{x}_i)) + \lambda\Omega(\|\Delta f\|_{\mathcal{H}}) + \beta\Psi(\|\mathbf{t}\|^2) \tag{24}$$

where $\Psi(\cdot)$ is a monotonically increasing regularization function, $\|\mathbf{t}\|^2$ is the norm of the weights, and β is a scalar. This new regularizer allows the framework to penalize large weights for auxiliary classifiers and favors small weights. This means the extended framework seeks to minimize the overall contribution of auxiliary classifiers, which is measured by $\|\mathbf{t}\|^2$. This may appear to be counter-intuitive given that this framework is used for adaptation, but it is reasonable when considered together with the other two goals of this framework.

The three terms in Eq.(24) indicates three factors to be minimized: (1) classification error, (2) the complexity of the delta function $\Delta f(\cdot)$, and (3) the contribution of the auxiliary classifiers $\{f_k^a(\cdot)\}_k$. So there is a contention between the contribution of $\Delta f(\cdot)$ and $\{f_k^a(\cdot)\}_k$. However, because the target classifier $f(\mathbf{x})$ is a combination of them (see Eq.(23)), they cannot be minimized at the same time or we will have $f(\mathbf{x}) \approx 0$ which cannot achieve small classification error. To reach a $f(\mathbf{x})$ with small classification error, one can either construct a relatively complex delta function, or construct a complex ensemble of auxiliary classifiers, or a combination of both approaches balanced by their respective cost measured by the two regularization terms. This framework prevents over-complicated delta function and too much reliance on auxiliary classifiers, as both of them are prone to overfitting.

5.1.2 Adaptive SVM with Domain Weighting (aSVM-DW)

As an extension of aSVM, we propose adaptive SVM with domain weighting, named as *aSVM-DW*, from the extended adaptation framework described above. This is realized by adopting SVM's hinge loss function $L(y, f(\mathbf{x}) = \max(0, 1 - yf(\mathbf{x})$ and using trivial regularization functions as $\Omega(x) = x$ and $\Psi(x) = x$. It is easy to show the objective function is equivalent to:

$$\min_{\mathbf{w},\mathbf{t}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}B\|\mathbf{t}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t. } \xi_i \geq 0, \quad y_i\sum_{t=1}^{M} t_k f_t^a(\mathbf{x}_i) + y_i\mathbf{w}^T\phi(\mathbf{x}_i) \geq 1 - \xi_i, \ \ \forall(\mathbf{x}_i, y_i) \in \mathcal{D}^t$$

By integrating the constraints through Lagrange multipliers, we can rewrite this objective function as a minimization problem of the following Lagrange (primal) function:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}B\|\mathbf{t}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}u_i\xi_i - \sum_{i=1}^{N}\alpha_i(y_i f(\mathbf{x}_i) - (1 - \xi_i)) \tag{25}$$

where $\alpha_i > 0$ and $u_i > 0$ are Lagrange multipliers. We minimize L_P by setting its derivative against $\mathbf{w}$, $\mathbf{t}$, and ξ to zero, which gives:

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \phi(\mathbf{x}_i), \qquad t_k = \frac{1}{B}\sum_{i=1}^{N}\alpha_i y_i f_k^a(\mathbf{x}_i), \qquad \alpha_i = C - \mu_i, \ \ \forall i \tag{26}$$

This equation on t_k shows a strong connection between weight t_k and the performance of the corresponding auxiliary classifier f_k^a on the target domain. Since $y_i f_k^a(\mathbf{x}_i)$

is a "margin" indicating how well f_k^a classifies instance $\mathbf{x}_i$, with larger margin indicating better performance, t_k as the weighted sum of the margins on all the examples in $\mathcal{D}^t$ measures the overall performance of f_k^a. This is intuitive because auxiliary classifiers with better performance usually lead to target classifiers with better performance by providing a higher starting point for adaptation. The extended adaptation framework and aSVM-DW embodies this intuitive principle. It justifies regularizer $\|\mathbf{t}\|^2$ in the objective function in Eq.(24) and in Eq.(25).

By plugging Eq.(26) into the primal Lagrangian Eq.(25), we obtain the dual Lagrange function as:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \Big(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{B} \sum_k f_k^a(\mathbf{x}_i) f_k^a(\mathbf{x}_j) \Big) \tag{27}$$

The parameters $\alpha = \{\alpha_i\}_{i=1}^N$ are estimated by maximizing L_D using another variation of SMO algorithm, which will be presented in Section 5.1.4. The target classifier expressed using the estimated $\hat{\alpha}$ is:

$$f(\mathbf{x}) \quad = \quad \sum_{i=1}^{N} \hat{\alpha}_i y_i \Big(K(\mathbf{x}_i, \mathbf{x}) + \frac{1}{B} \sum_{k=1}^{M} f_k^a(\mathbf{x}_i) f_k^a(\mathbf{x}) \Big) \tag{28}$$

5.1.3 Discussion of aSVM-DW

We offer an alternative interpretation of aSVM-DW as a normal SVM model that treats *the outputs of auxiliary classifiers as additional features*. For each instance $\mathbf{x}$, we treat the outputs of auxiliary classifiers $\mathbf{f} = [f_1^a(\mathbf{x}), ..., f_M^a(\mathbf{x})]$ as an extra feature vector in addition to its original feature vector $\mathbf{x}$. Similarly, $\mathbf{f}_i = [f_1^a(\mathbf{x}_i), ..., f_M^a(\mathbf{x}_i)]$ is an additional feature vector to $\mathbf{x}_i$. We can rewrite the target classifier in Eq.(28) as $f(\mathbf{x}) = \sum_i \alpha_i y_i (K(\mathbf{x}_i, \mathbf{x}) + \frac{1}{B}\mathbf{f}_i \cdot \mathbf{f})$. While $K(\mathbf{x}_i, \mathbf{x})$ is the similarity between $\mathbf{x}_i$ and $\mathbf{x}$ in the (transformed) feature space, and $\mathbf{f}_i \cdot \mathbf{f}$ measures the similarity of two instances in terms of the outputs of auxiliary classifiers on them. If the classifiers' outputs on them are close, in which case $\mathbf{f}_i \cdot \mathbf{f}$ is large, their similarity is high, and vice versa. Compared with a SVM classifier $f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})$, an aSVM-DW classifier simply extends the measure of similarity to include the similarity on the classifier-output space.

In the linear case where a trivial kernel function $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i \cdot \mathbf{x}$ is used, we can concatenate feature $\mathbf{x}$ with classifiers outputs $\mathbf{f}$ to form a "hybrid" feature vector $\mathbf{z} = [\begin{smallmatrix} \mathbf{x} \\ \mathbf{f} \end{smallmatrix}]$, and similarly, $\mathbf{z}_i = [\begin{smallmatrix} \mathbf{x}_i \\ \mathbf{f}_i \end{smallmatrix}]$. Assuming $B = 1$ without loss of generality, the target classifier in aSVM-DW can be written as $f(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{z}_i \cdot \mathbf{z}$, which is identical to a linear SVM model based on the concatenated feature vector $\mathbf{z}$.

This "feature concatenation" view does not apply in general cases where $K(\cdot, \cdot)$ is non-trivial. This is because the feature similarity is computed in any kernel space, while the classifier-output similarity is computed in linear space. Therefore, we cannot implement aSVM-DW as SVM over concatenated feature vectors. Neither can we use the training algorithm for SVM to learn aSVM-DW by simply replacing $K(\mathbf{x}_i, \mathbf{x}_j)$ with $K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{B}\mathbf{f}_i \cdot \mathbf{f}_j$. We will explore this issue in Section 5.1.4.

The idea of treating model (classifier) outputs as additional features is not new. For example, Natsev et al. [56] adopted the so called "model-vector" approach which constructed semantic feature of a video shot from the scores of video concept classifiers and used it in retrieval. Wu et al. [89] used the outputs of concept classifiers built on different modalities as meta-features for concept detection. Here, we offer a more principled interpretation of this seemingly ad-hoc technique: using classifier outputs as features is equivalent to adapting these classifiers under the regularized loss minimization framework.

The above discussion provides insights on the scalar B. We see the role of B as to balances the contribution between feature similarity and the similarity based on auxiliary classifier outputs. The scale of these two similarity terms is affected by the feature dimension d and the number of auxiliary classifiers M, respectively. Typically, we have $d \gg M$ since features for multimedia data is of high dimension. We need to set $\frac{1}{B} > 1$ to avoid the classifier-output similarity being overwhelmed by the feature similarity. A good starting point is to set $B = \frac{M}{d}$ such that that the two similarity terms have equal contribution.

5.1.4 Learning Algorithm for aSVM-DW

The fact that the dual form of aSVM-DW (in Eq.(27)) can be derived from SVM's dual form by replacing $K(\mathbf{x}_i, \mathbf{x}_j)$ with $K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{B}\sum_k f_k^a(\mathbf{x}_i) f_k^a(\mathbf{x})$ gives the impression that aSVM-DW can be implemented using SVM's learning algorithm by replacing the kernel term. This impression is not true. Despite the correspondence between their dual form, the intercept term of a classifier is involved in the objective function of aSVM-DW through regularizer $\|\mathbf{w}\|^2$, but it is not involved in the objective function of SVM. Based on our discussion in Section 4.3.3, this allows the SMO algorithm for aSVM-DW to optimize only one working variable in each iteration, instead of two variables as for SVM. This changes the structure of the learning algorithm.

The SMO algorithm for aSVM-DW can be modified from that for aSVM we described in Section 4.3.3. In fact, the optimality condition and selection criterion of working variable remain exactly the same as for aSVM, which are given by Eq.(11) and Eq.(12). The only difference is the analytical solution of each working variable, because aSVM-DW has a different optimization function. Suppose α_1 is the working variable. We set the derivative of the dual form in Eq.(27) against it to zero:

$$\frac{\partial L_D}{\partial \alpha_1} = 1 - y_1 f^{old}(\mathbf{x}_1) - (\alpha_1^{new} - \alpha_1^{old})\left(K(\mathbf{x}_1, \mathbf{x}_1) + \frac{1}{B}\sum_{k=1}^{M} f_k^a(\mathbf{x}_1)^2\right) \triangleq 0 \qquad (29)$$

which leads to the analytical solution of α_1:

$$\alpha_1^{new} = \alpha_1^{old} + \frac{1 - y_1 f^{old}(\mathbf{x}_1)}{K(\mathbf{x}_1, \mathbf{x}_1) + \frac{1}{B}\sum_{k=1}^{M} f_k^a(\mathbf{x}_1)^2} \qquad (30)$$

This optimal solution may need to be clipped to satisfy the constraint $0 < \alpha_1 < C$. Thus, the learning algorithm of aSVM-DW is the same as that for aSVM except that the variable update equation in Eq.(13) needs to be replaced by Eq.(30).

5.2 Extraction of Domain Features

The approach in the previous section associates the weights of auxiliary classifiers with their classification performance on the labeled examples in the target domain. The underlying assumption is that auxiliary classifiers perform well on the target domain is more useful in adaptation. While this is a valid assumption, a classifier's performance on labeled examples $\mathcal{D}^t$ is not a reliable approximation of its performance on the whole domain, because $\mathcal{D}^t$ is too small to faithfully represent the distribution of domain population. Labeling more examples in the target domain may improve such approximation but would offset the benefit of adaptation on minimizing labeling effort. In order to better weight auxiliary domains, we explore several *domain features* that indicate how closely different domains are related or how well a classifier trained from one domain performs on another domain. These features are combined to predict the performance of an auxiliary classifier on the target domain. We discuss the extraction of several domain features below.

5.2.1 Domain Metadata

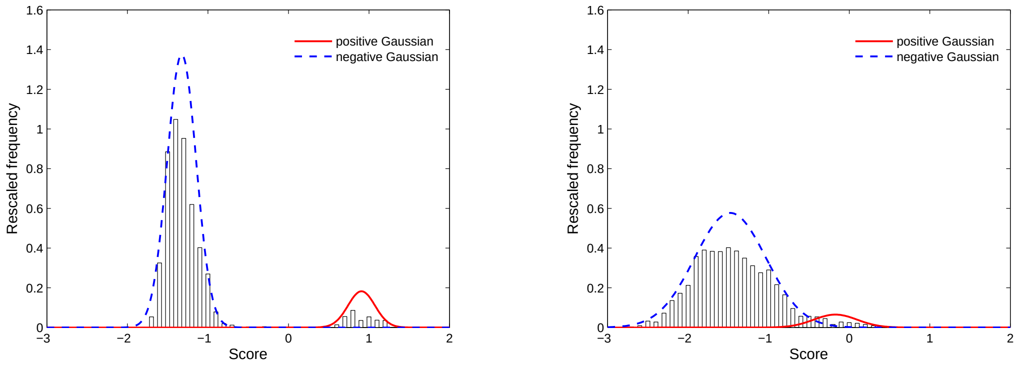
In many cases, metadata are available which describe domains in terms of the genre, style, editor and provider of the data. These metadata provide important clues as to how much two domains are related to each other. For example, in TRECVID 2005 corpus, the language, country, and broadcaster (channel) information of news video footage is available. It is reasonable to assume the footage edited by broadcaster in the same country, such as CNN and NBC data, are more related than footage from broadcasters of two countries, such as CNN and CCTV. We may also assume that the relatedness between TREC05 data and TREC07 data is weaker because the latter are documentary video data.

5.2.2 Features on Score Distribution

A classifier produces a score on each data instance to indicate how likely its label is a positive (or negative). When a classifier is applied to a dataset, the distribution of the scores provide clues as to how well this classifier performs. For a "good" classifier, the scores of positive instances and those of negative ones are well separated, while for a "poor" classifiers their scores are mixed together. This between-class score separation is a good indicator of the classifier's performance, but it requires the data labels which are unavailable in a target domain. We propose a model-based approach to overcome this problem. Assuming the scores of positive and negative data follow distributions of a certain family, we recover their respective distributions from classifier outputs using Expectation Maximization (EM) algorithm [37] in order to extract features like score separation.

Models of score distribution have been explored for combining multiple search engine outputs [51] and rank aggregation [25]. Given the diversity of classification algorithms, it is impossible to find a distribution family that fits the scores produced by any classifier. Thus, we made several assumptions on classifier output: (1) the scores are in the range of $(-\infty, +\infty)$; (2) the boundary between two classes is zero, with scores above zero indicating positive instances, and vice versa; (3) the absolute value of a score indicates the degree of confidence on the predicted label. While seemingly restrictive, these assumptions are actually general enough to include widely used classification methods in multimedia, such as SVM and (kernel) logistic regression.

Based on these assumptions, we use Gaussian distributions to fit the scores of pos-



(a) Classifier A (AP=0.972)

(b) Classifier B (AP=0.124)

Figure 9: The score distribution of two classifiers on the same dataset. The histograms show the actual score distributions, and the Gaussian curves are fit by EM. Classifier A performs better than B.

itive instances and of negative instances respectively, because they fit the actual scores well and their parameters are easy to estimate. Suppose $z = f^a(\mathbf{x})$ is the score that auxiliary classifier f^a produces on instance $\mathbf{x}$. The scores of positive instances follow distribution $p(z|y = 1) = \mathcal{N}(u_p, \sigma_p^2)$, where u_p and σ_p^2 are the mean and variance. Similarly, $p(z|y = -1) = \mathcal{N}(u_n, \sigma_n)$ is the distribution of the scores of negative instances. We assume the prior of labels to be $P(y = 1) = \pi$ and $P(y = -1) = 1 - \pi$. The overall score distribution is therefore a Gaussian mixture model with two components:

$$p(z) = \pi\mathcal{N}(u_p, \sigma_p) + (1 - \pi)\mathcal{N}(u_n, \sigma_n) \tag{31}$$

The parameters $(\pi, u_p, \sigma_p, u_n, \sigma_n)$ are estimated from the outputs of $f^a(\mathbf{x})$ on all (labeled and unlabeled) the data in the target domain using the EM algorithm. It iteratively improves the model parameters until two Gaussian components that best fit the scores are found. Figure 9 shows the estimated score distributions of two "studio" classifiers, one trained from a NTDTV news program and the other from a NBC news program in TRECVID 2005 corpus, being applied to another NTDTV program (see details of these data in Section 3.1). We find that the Gaussian mixture models estimated by EM fit the score distributions well. Moreover, it shows a strong relationship between the score distribution and the classifier performance, because the classifier with larger between-class score separation has much higher performance.

We extract various features from the estimated score distributions. Suppose $(\hat{\pi}, \hat{u}_p, \hat{\sigma}_p, \hat{u}_n, \hat{\sigma}_n)$ are the parameters estimated by EM. One feature describing the score separation is the distance between the mean of the positive and negative data as $\hat{u}_p - \hat{u}_n$. A slightly different feature, which is proven to be effective in previous work [51], is the distance between the mean of the positive $\hat{u}_p$ to the middle point z^{mid} at which the two Gaussian densities intersect (i.e., $p(z^{mid}|y = 1) = p(z^{mid}|y = -1)$).

Feature	**Description**
sample_ap	average precision of f^a on the labeled subset $\mathcal{D}^t$ of the primary data
max_score	the maximum score produced by f^a on the primary data
pos_neg_dist	distance b/w the estimated mean scores of positive data and of negative data
pos_mid_dist	distance b/w the mean score of positive data to the point where the positive and negative score distribution intersects
pseudo_ap	average precision of f^a computed on pseudo labels derived from score aggregation

Table 5: Domain features for predicting the performance of an auxiliary classifier $f^a(\mathbf{x})$.

5.2.3 Pseudo Performance

Although most data in the target domain are unlabeled, we can evaluate auxiliary classifiers based on the "pseudo labels" on the target data. One way to compute the pseudo labels is based on the notion that the "average" of multiple classifiers is usually better than any individual classifier. Therefore, we aggregate the outputs of multiple auxiliary classifiers to predict the labels of the primary data, and then use these pseudo labels to evaluate each classifier. Assuming $z_i^1, ..., z_i^M$ are the scores on instance $\mathbf{x}_i$ produced by a set of M auxiliary classifiers, a principled way to aggregate these scores is to compute the posterior distribution as $P(y_i = 1|z_i^1, ..., z_i^M)$. If the outputs of different classifiers are independent given an instance and its label, we have the following based on Baye's rule:

$$P(y_i = 1|z_i^1, ..., z_i^M) = \frac{P(y_i = 1)\prod_{k=1}^M p(z_i^k|y_i = 1)}{\sum_{y_i=-1,1} P(y_i)\prod_{k=1}^M p(z_i^k|y_i)} \tag{32}$$

where $p(z_i^k|y_i = 1)$ and $p(z_i^k|y_i = -1)$ are Gaussian distributions fit by the EM algorithm described above, and the prior $P(y_i)$ is set to the ratio of positive (or negative) instances in the training data of these classifiers.

Each individual classifier is evaluated by measuring the agreement between its output and the estimated posterior probability. One way is to convert the posteriors into the pseudo labels as $\hat{y}_i = sgn(P(y_i = 1|z_i^1, ..., z_i^M) - 0.5)$, and compute a certain performance metric (e.g., average precision (AP)) of the classifier based on the pseudo labels. Another way is to convert the posteriors into ranks and measure the consistency between this aggregated rank and the rank generated by a classifier using the Kendall tau distance. Empirically we find the measure based on pseudo labels is better.

5.2.4 Predicting Classifier Performance

We build a regression model to predict an auxiliary classifier's performance on the target domain based on the domain features. Table 5 summarizes the domain features used as input to the regression model, including domain metadata, features computed from score distribution and pseudo performance. The input also includes the classifier's performance on the labeled examples measured by average precision (AP). The output of the regression model is the predicted performance (also in terms of AP) of this classifier on the whole target domain. The regression model is trained using support

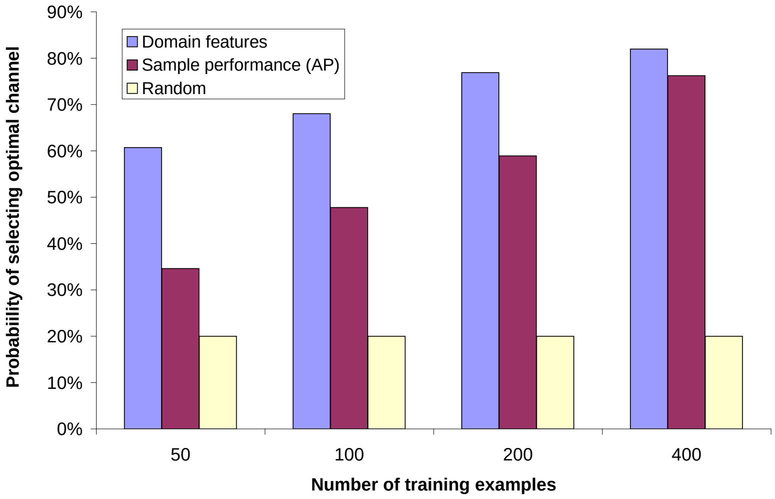


Figure 10: The probability of selecting optimal auxiliary channel for channel CNN on 39 concepts.

vector regression (SVR) [26]. We then select and weight the auxiliary classifiers based on their performed predicted by this model.

5.2.5 Preliminary experiment

We evaluate the use of domain features for selecting auxiliary classifiers (domains) in cross-channel video concept detection. Specifically, we use CNN to be the target channel, and rank the other 5 channels (CCTV, NBC, MSNBC, NTDTV, and LBC) as candidate auxiliary channels in terms of their utility using the following criteria:

- `Random` ranks the auxiliary channels randomly.
- `Sample performance` ranks the auxiliary channels by the performance of their corresponding classifiers based on the labeled examples in the target channel.
- `Domain feature` ranks the auxiliary channels according to the predicted performance based on the domain features, with the approach described in the previous subsection.

For each of the 38 concepts, we rank the 5 channels using one of the above criteria, and choose the top-ranked one to be the only auxiliary channel. On the other hand, we evaluate the classifiers of these 5 channels based on the labels of CNN data, and the one with the highest *actual* performance is the optimal choice. We evaluate each selection criterion in terms of the probability that the top-ranked channel is indeed the optimal one. Figure 10 summarizes the evaluation results at different sample size. It shows that using domain features significantly outperforms the selection made according to the performance on labeled examples. Also, the selection of both methods improves as more labeled examples are available.

5.3 Proposed Work: Adaptation with Domain Features

Section 5.1 introduces an extended framework capable of automatic weighting of auxiliary classifiers, where the weights are associated with their performance on the labeled examples. Section 5.2 presents a set of domain features which indicate domain relatedness and utility of auxiliary classifiers. Since both approaches exploit orthogonal knowledge, it is advantageous to combine them.

We plan to further extend the adaptation approach in section 5.1 such that the weights of auxiliary classifiers are influenced by the domain features. Suppose the domain features of domain k are denoted as $\mathbf{z}_k \in \mathbb{R}^l$ where l is the dimension, and $\mathbf{z} = [\mathbf{z}_1^T, .., \mathbf{z}_M^T]^T$ is a $l \times M$ matrix stacked from the domain feature vectors. We assume the weight of auxiliary classifier follows a Gaussian distribution with mean parameterized by a linear combination of domain features, i.e., $t_k \sim \mathcal{N}(\mathbf{u}^T\mathbf{z}_k, \sigma^2)$, where $\mathbf{u} = \{u_k\}_{k=1}^l$ is a set of l parameters, and σ is the predefined standard deviation. We modify the regularized loss function as:

$$\min_{\Delta f, \mathbf{t}, \mathbf{u}} \sum_i L(y_i, f(\mathbf{x}_i)) + \lambda\Omega(\|\Delta f\|_{\mathcal{H}}) + \beta\Psi(\|\mathbf{t} - \gamma^T\mathbf{z}\|) \tag{33}$$

Under the hinge loss function of SVM, we derive adaptive SVM with domain feature based weighting (aSVM-DFW) from this general framework. The objective function of aSVM-DFW is written as:

$$\min_{\mathbf{w}, \mathbf{t}, \mathbf{u}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{B}{2}\|\mathbf{t} - \mathbf{u}^T\mathbf{z}\|^2 + C\sum_{i=1}^{N}(1 - y_i f(\mathbf{x}_i))_+ \tag{34}$$

where $f(\mathbf{x}) = \sum_{k=1}^M t_k f_k^a(\mathbf{x}) + \Delta f(\mathbf{x})$. aSVM-DFW can be seen as a step further beyond aSVM-DW by integrating the domain features $\mathbf{z}_k$. The learning algorithm of aSVM-DFW and its performance will be explored in this thesis.

6 Sample Selection for Domain Adaptation

The previous sections have covered the problems of adapting auxiliary classifiers to a target domain as well as the selection and weighting of auxiliary classifiers. One key problem that is raised in Section 1.3 but has not been addressed is how to select training examples from the target domain to help the adaptation process. So far the training examples have been randomly, which is clearly suboptimal. We assume that adaptation based on *informative* samples will either lead to better classifiers for the target domain, or require fewer examples to reach the same classification accuracy. So effective sample selection methods are important to improving classification performance and reducing labeling effort.

The problem of sample selection has been extensively studied in research on *active learning* [19], but no methods are devoted to the context of adaptation. In previous work, a number of methods have been proposed for selecting informative samples for large-margin classifiers, especially SVM [12, 70, 83, 93]. However, all these methods consider building *new* classifiers from the selected examples, while we consider adapting *existing* classifiers based on these samples. The difference on context makes a difference in sample selection strategy. Instead of selecting samples that are informative for building classifiers from scratch, we select samples that provide information *complementary* to auxiliary classifiers so that they can be adapted to target classifiers with better performance using fewer examples. Said in another way, we need to make sure the information provided by examples is not redundant with the information of auxiliary classifiers.

We describe two sample selection methods in the context of classifier adaptation, which represent two different ways of finding examples with complementary information. The first method selects examples to achieve the largest loss reduction on auxiliary classifiers, while the second one selects examples causing maximal disagreement between auxiliary classifiers. We describe their details and evaluate their performance in this section.

6.1 Sample Selection based on Loss Minimization

6.1.1 The Theoretical Model

Formally, let $\mathcal{P}$ be a pool of data in the target domain, and $\mathcal{D}$ be the set of examples selected from $\mathcal{P}$ to label. Also, let $P(y|\mathbf{x})$ be the conditional probability of the label of an example $\mathbf{x}$ in $\mathcal{P}$, and $P(\mathbf{x})$ be the marginal probability of $\mathbf{x}$. Following the formulation of regularized loss in previous sections, we use $L(y, f(\mathbf{x}))$ to denote the loss of the target classifier f on instance $\mathbf{x}$, and $Reg(f)$ as the regularization term that is related to the complexity of $f(\mathbf{x})$. For example, the regularizer is $\Omega(\|\Delta f\|_{\mathcal{H}})$ in the adaptation framework in Section 4.2. We express the *expected risk function* as follows:

$$\begin{aligned} R(f) &= E_{\mathbf{x}} E_{y|\mathbf{x}}(L(y, f(\mathbf{x})) + Reg(f) \\ &= \int_{\mathcal{X}} \sum_{y \in \mathcal{Y}} L(y, f(\mathbf{x})) P(y|\mathbf{x}) P(\mathbf{x}) d\mathbf{x} + Reg(f) \end{aligned} \quad (35)$$

Given the difficulty of computing the expected risk over the distribution $P(\mathbf{x})$, it is more feasible to measure this risk over the pool of available data $\mathcal{P}$:

$$R(f) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{x} \in \mathcal{P}} E_{y|\mathbf{x}}(L(y, f(\mathbf{x})) + Reg(f) \tag{36}$$

For simplicity, we assume there is only one auxiliary classifier f^a. Suppose $f^{\mathcal{D}}$ is the target classifier adapted from f^a based on labeled examples in $\mathcal{D}$ using any adaptation method proposed in Section 4. During the adaptation process, $f^{\mathcal{D}}$ is initially equal to f^a when $D = \varnothing$, and it gradually deviates from f^a when more examples are included in $\mathcal{D}$. In this case, the optimal set of samples $\mathcal{D}_{opt}$ are those that minimize the expected risk of the target classifier $f^{\mathcal{D}}$, i.e., $\mathcal{D}_{opt} = \mathrm{argmin}_{\mathcal{D}} R(f^{\mathcal{D}})$. This is equivalent to the sample set that achieves the *largest risk reduction* from the auxiliary classifier f^a:

$$\begin{aligned} \mathcal{D}_{opt} &= \underset{\mathcal{D}}{\mathrm{argmax}} \Big(R(f^a) - R(f^{\mathcal{D}}) \Big) \qquad (37) \\ &= \underset{\mathcal{D}}{\mathrm{argmax}} \sum_{\mathbf{x} \in \mathcal{P}} E_{y|\mathbf{x}} \Big(L(y, f^a(\mathbf{x})) - L(y, f^{\mathcal{D}}(\mathbf{x})) \Big) + \Big(Reg(f^a) - Reg(f^{\mathcal{D}}) \Big) \end{aligned}$$

6.1.2 Computationally Tractable Methods

Theoretically, directly maximizing Eq.(37) leads to the optimal sample set. In practice, however, this is prohibitively expensive because there is $2^{|\mathcal{P}|}$ possible choices for $\mathcal{D}$ and for each choice $f^{\mathcal{D}}$ needs to be re-trained to update the estimate of the expected loss function. In the following, we introduce several simplifying assumptions which lead to two computationally tractable sample selection methods.

First, we use the risk reduction over the sample set $\mathcal{D}$ to approximate the risk reduction over the whole collection $\mathcal{P}$. This can be done by replacing the range of the sum in Eq.(37) from $\mathbf{x} \in \mathcal{P}$ with $\mathbf{x} \in \mathcal{D}$. The underlying assumption is that the expected loss for any $\mathbf{x} \in \mathcal{P} \setminus D$ has an equal influence. The same approximation was made in previous active learning methods such as [12] and [93].

Moreover, we assume that the target classifier $f^{\mathcal{D}}$ can correctly predict the labels of any $\mathbf{x} \in D$. This suggests the risk $\sum_{\mathcal{D}} E_{y|\mathbf{x}} L(y, f^{\mathcal{D}}(\mathbf{x}))$ is negligible compared with $\sum_{\mathcal{D}} E_{y|\mathbf{x}} L(y, f^a(\mathbf{x}))$ so that it can be dropped. This is a reasonable approximation because a kernel machine such as (adaptive) SVM can easily find a non-linear decision boundary to separate a small number of training examples by their classes. Based on these two assumptions, we rewrite the expression of $\mathcal{D}_{opt}$ as follows:

$$\mathcal{D}_{opt} = \underset{\mathcal{D}}{\mathrm{argmax}} \sum_{\mathbf{x} \in \mathcal{D}} E_{y|\mathbf{x}} L(y, f^a(\mathbf{x})) + \Big(Reg(f^a) - Reg(f^{\mathcal{D}}) \Big)$$

Compared with Eq.(37), this has simplified the problem, yet it is still computationally intractable because $f^{\mathcal{D}}$ needs to be trained for every choice of $\mathcal{D}$. So we make two further assumptions. First, we assume the samples are picked in a “greedy” manner. That is, we repeatedly pick the single best sample from the pool of un-sampled instances, and accumulate them into a sample set. This will not lead to the optimal sample set, but it reduces the number of choices of $\mathcal{D}$ from $2^{|\mathcal{P}|}$ to $|\mathcal{P}|$ in each iteration. Based on this strategy, we further assume that the change of regularization term as $Reg(f^a) - Reg(f^{\mathcal{D}})$ after each sample is included into $\mathcal{D}$ does not depend on the choice of this sample. This certainly not true, because $f^{\mathcal{D}}$ depends on the choice of samples. However, the change of the regularizer for one sample is typically very small and hard to estimate directly from the sample without re-training $f^{\mathcal{D}}$, which we want to avoid. With these further assumptions, our sample selection strategy is equivalent to repeatedly choosing the best sample $\mathbf{x}_{opt}$ using the following criterion and add it to the existing sample set, i.e., $D = D \bigcup \{\mathbf{x}_{opt}\}$:

$$\mathbf{x}_{opt} = \underset{\mathbf{x} \in \mathcal{P} \backslash \mathcal{D}}{\operatorname{argmax}} \sum_{y=\{-1,1\}} P(y|\mathbf{x}) L(y, f^a(\mathbf{x})) \tag{38}$$

Eq.(38) eliminates the computationally intensive step of re-training f^D for every choice of $\mathcal{D}$, and reduces the search space from $2^{|\mathcal{P}|}$ to $|\mathcal{P}|$. This allows the best sample to be found very efficiently. The rationale behind Eq.(38) is simple and intuitive: find the example with the largest loss under f^a so that by including it into training set we can reduce the overall loss the most. In another words, if an example is the one that f^a is *least likely* to correctly classify, then it provides knowledge *most complementary* to the auxiliary classifiers.

Finally, we need to estimate the conditional probability $P(y|\mathbf{x})$. Although the classifier outputs a confidence score $f(\mathbf{x})$ which indicates the relevance of each $\mathbf{x}$ to the target class, it is not a reliable estimation of the posterior probability, even if it is normalized to $[0, 1]$. Therefore we develop two models for estimate $P(y|\mathbf{x})$, both of which are based on the hinge loss function of SVM, i.e., $L(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))$, because the same loss function is used in aSVM and its variations as well.

- **Prior Model:** We assume the class-conditional $P(y|x)$ is completely *unrelated* to the prediction made by f^a; instead, we set it to the prior distribution of label y in the data of auxiliary domain $\mathcal{D}^a$ if available, i.e., $P(y = \pm 1|x) \approx P_{\mathcal{D}^a}(y = \pm 1)$, where $P_{\mathcal{D}^a}(y = \pm 1)$ is the ratio of positive or negative instances in $\mathcal{D}^a$, respectively. In this case, the sample selection criterion in Eq.(38) is written as:

$$\mathbf{x}_{opt} = \underset{\mathbf{x} \in \mathcal{P} \backslash \mathcal{D}}{\operatorname{argmax}} \sum_{y=\{1,-1\}} \Big(P_{\mathcal{D}^a}(y) \max(0, 1 - yf^a(\mathbf{x})) \Big) \tag{39}$$

  This model is related to the "Uniform Guess" model in [93] and "Random Labels" model in [12] since they both dissociate $p(y|\mathbf{x})$ with $f(\mathbf{x})$ and set it uniformly (to 1/2). Using the prior distribution in the old data, our model is more accurate when the two classes are unbalanced, which is typical in multimedia problems. When the positive data are rare, i.e., $P_{\mathcal{D}^a}(y = 1) \ll P_{\mathcal{D}^a}(y = -1)$, the selection criterion in Eq.(39) is biased towards selecting the examples considered "most positive" by f^a. This is desired since the examples from the rare class are more useful in terms of training a classifier.

- **Best Worst Model:** This model has been suggested in several papers [12, 93] and it leads to the well-known "uncertainty sampling" strategy [83]. In this model, we approximate the expected loss with the smallest loss among all the possible labels. Because the smallest loss always comes with the predicted label, this model assumes that f^a correctly predicts the label of $\mathbf{x}$, i.e., $y = sgn(f^a(\mathbf{x}))$. In this case, Eq.(38) is written as:

$$\begin{aligned}
\mathbf{x}_{opt} &= \operatorname*{argmax}_{\mathbf{x}\in\mathcal{P}\backslash D}\left\{\min_{y=-1,1}\Big(\max(0, 1-yf^a(\mathbf{x}))\Big)\right\} \\
&= \operatorname*{argmax}_{\mathbf{x}\in\mathcal{P}\backslash D}\left\{\max(0, 1-sgn(f^a(\mathbf{x}))f^a(\mathbf{x}))\right\} \\
&= \operatorname*{argmax}_{\mathbf{x}\in\mathcal{P}\backslash D}\left\{\max(0, 1-|f^a(\mathbf{x})|)\right\} \\
&= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{P}\backslash D}|f^a(\mathbf{x})| \qquad (40)
\end{aligned}$$

Since $|f^a(\mathbf{x})|$ indicates how confidence the classifier is about the predicted label, this sample selection strategy chooses the examples that f^a is least confident about, i.e., the most ambiguous examples. They are also the examples that are closest to the decision boundary which is specified by $f^a(\mathbf{x}) = 0$. This is similar to the uncertainty sampling strategy in active learning [12, 83] except that the decision is made based on the auxiliary classifier f^a.

6.1.3 Batch and Incremental Sample Selection

The sample selection methods described above can be used in two modes: batch mode or incremental mode. In the batch model, we collect a set of samples by repeatedly selecting the best sample, label all of them, and adapt the auxiliary classifier to the target classifier based on the labeled samples. The learning (adaptation) is done only once, i.e., after all the samples are collected and labeled. No learning takes place based on part of the samples when they are still being collected. In this case, all the samples are selected based on the risk function (see Eq.(38)) of f^a, which is efficient because f^a is given.

In contrast, the incremental mode requires the target classifier to be repeatedly adapted whenever a new sample(s) arrives. The new classifier is adapted from the classifier trained in the previous round based on the most recent sample(s). This means the new classifier, once trained, immediately becomes the auxiliary classifier for the next round of adaptation. In this mode, samples are selected based on the loss function of the most recent target classifier $f^{\mathcal{D}}$, so they are supposed to be more effective (in terms of risk reduction). This mode is useful in interactive systems where there is a need to repeatedly update target classifiers and show updated results while users are selecting and labeling new examples.

6.1.4 Sample Selection with Multiple Auxiliary Classifiers

The aforementioned sample selection methods assume only one auxiliary classifier. To extend it to the case of multiple auxiliary classifiers $f^a_1, ..., f^a_M$, we extend the sample selection criterion in Eq.(38) to find the example with the largest *average* loss over all the auxiliary classifiers, or the example with the largest *maximum* loss over all the auxiliary classifiers:

$$\begin{aligned} & \mathbf{x}_{opt} = \underset{\mathbf{x}\in\mathcal{P}\backslash\mathcal{D}}{\operatorname{argmax}} \sum_{k=1}^{M} \sum_{y=\{-1,1\}} P(y|\mathbf{x}) L(y, f_k^a(\mathbf{x})) \\ \text{or} \quad & \mathbf{x}_{opt} = \underset{\mathbf{x}\in\mathcal{P}\backslash\mathcal{D}}{\operatorname{argmax}} \max_{k=1,..,M} \sum_{y=\{-1,1\}} P(y|\mathbf{x}) L(y, f_k^a(\mathbf{x})) \end{aligned} \tag{41}$$

The downside of this sample selection method is that it treats the auxiliary classifiers equally while in reality they are not equally useful. Ideally, one should give emphasis on the loss function of "useful" auxiliary classifiers. However, the weights indicating the usefulness of auxiliary classifiers are not learned until the target classifier is trained. Therefore, if we have the weights as the result of previous adaptation iterations, we can exploit them in the sampling process. Otherwise, the best we can do is to treat the auxiliary classifiers equally.

6.2 Proposed Work: Sample Selection based on Classifier Disagreement

Another set of active learning methods [53, 71] seek to find examples that cause *maximal disagreement* across a set of classifiers on the same task, i.e., examples that half of the existing classifiers classify as positive and the other half classify as negative. The underlying theory is to find examples that help reduce the version space the most, where version space is the subset of all hypotheses that are consistent with the current set of labeled examples. Methods such as Query-by-Committee (QBC) by Seung et al. [71] suggest to train a committee of classifier variants to represent the range of the version space. Examples causing maximal disagreement can invalidate half of the classifiers and thus reduce the version space approximately by half. Tong and Koller [82] also show that the uncertainty sampling method in SVM has a theoretical foundation in maximal reduction of version space.

We borrow the principle of maximal disagreement for identifying examples with multiple auxiliary classifiers. Instead of training a set of classifiers using a randomized learning algorithm, we can use the set of existing auxiliary classifiers to identify examples that cause maximal disagreement among them.

6.3 Preliminary Experiments on Sample Selection

We compare the two sample selection methods derived from loss minimization framework 6.1, which are based on the Prior model and the Best-Word model, with random sampling in cross-channel video concept detection. We use the same experiment setting used in Section 4.5. To classify video shots in channel CNN, we adapt concept classifiers trained from channel NBC using aSVM based on examples selected by different selection methods.

Figure 11 compare the performance of aSVM using different sample selection methods in terms of MAP averaged from 38 concepts. We find that the two loss-reduction sampling methods based on Prior model (`prior`) and based on Best-Word model (`best-worst`) are considerably better than random sampling (`random`). This shows that using

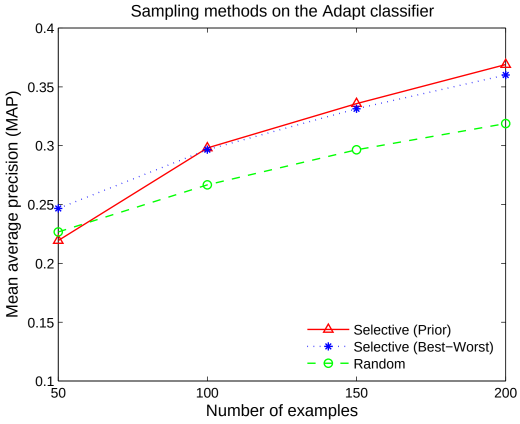


Figure 11: Comparison of sample selection methods based on aSVM.

informative samples instead of random samples does improve the performance of the target classifier derived from adaptation. Between the two sample selection methods, `prior` is slightly better than `best-Worst` except when there are only 50 samples. This suggests the auxiliary classifier does not do a better job on predicting the data labels than guessing according to the ratio of the two classes.

7 Proposed Work: Domain Adaptation in Other Applications

We have demonstrated that the success of the proposed adaptation approaches in solving the problem of cross-domain video concept detection. Nevertheless, our approach is general enough to be applied to many other adaptations within and beyond the area of multimedia, as long as they can be formulated as classification problems. We plan to investigate two specific problems in the thesis work:

- **Adaptation of retrieval models**: Multimedia retrieval models combine the scores of various “retrieval experts”, such as text retrieval score, image similarity, as a weighted sum into a final score indicating the relevance of an image or video shot to a user query. The combination weights are learned using a linear classifier (e.g., logistic regression) trained from past queries with labeled results for each query. A retrieval model can be built for a certain corpus (e.g., CNN news video), a certain type of queries (e.g., queries for people), or even every query, depending the availability of training data. There is a need to adapt the retrieval model(s) when the underlying corpus changes, when the type of queries changes, or even when there is a unseen query. In this work, we can treat existing retrieval models as auxiliary classifiers, the new corpus or query type or query as the target domain, and (pseudo) relevant examples obtained through implicit or explicit user feedbacks as training examples. The goal is to build an effective retrieval models by adapting from existing ones without having to label many training examples or not labeling any at all.
- **Cross-corpora text categorization**: There is a large variety in terms of the text corpus used for text categorization. Popular corpora include the Wall Street Journal (WSJ), 20-news group data, and so on. To our knowledge, no prior work has been done on the problem of applying classifiers trained from one corpus to other corpora. Through experiments, we will study how much corpus change affects the performance of text classification, and how much our adaptation approaches help to alleviate the impact. This will show the effectiveness of our approach high-dimensional but discrete term-vector features used in text documents.

In addition to these two applications, we will seek other opportunities of applying our approaches to solve adaptation problems in various research areas.

8 Summary of Proposed Thesis Work

8.1 Domain Impact to Video Concept Detection

We will conduct a comprehensive empirical study on the impact of domain change to the performance of video concept detection. This can be realized by building concept classifiers from one domain and examining their performance on the other domains. The test data used include TRECVID corpus in 2005 through 2007, which include broadcast news video and documentary video from different programs and channels. By varying the choices of "training domain" and "test domain", we will study how robust the concept classifiers are over the change of programs, content providers (channels), video types (news and documentary), by comparing the classification performance on in-domain data and on out-of-domain data. This study will also provide insights on the robustness or domain-dependency of different types of low-level features and different concepts.

8.2 Domain Analysis

- We will implement *adaptive SVM with domain weighting* (aSVM-DW) based on its formulation and learning algorithm presented in Section 5.1. This algorithm is able to adapt a set of auxiliary classifiers to a single target classifier and automatically learn the weights of these auxiliary classifiers in the same process.
- A limitation with aSVM-DW is that the weights of auxiliary classifiers are only determined by their performance on classifying a small number of examples in the target domain. To learn more reliable weights, we will further extend aSVM-DW to integrate the domain features described in Section 5.3 which indicates the relatedness of auxiliary classifiers (domains) to the target domain. The extended algorithm will be named adaptive SVM with domain feature based weighting (aSVM-DFW).
- We will evaluate the performance of various domain analysis approaches, including aSVM-DW and aSVM-DFW, in cross-domain video concept detection. Besides comparing different approaches, we plan to investigate several issues through the experiment, including the difference between using "good" and "bad" auxiliary classifiers, impact of the number of auxiliary classifiers, etc.

8.3 Sample Selection for Adaptation

- We will implement a sample selection method based on the "maximal disagreement" principle, which intends to find examples in the target domain whose label is predicted as positive by half of the auxiliary classifiers and negative by the other half. The underlying idea is to find examples that help reduce the version space the most.
- We plan to extend the current preliminary experiment on sample selection methods in video concept detection. The new experiment will include the maximal-disagreement sample selection method besides the methods based on loss minimization, and as well as settings with multiple auxiliary classifiers. In order to

provide deeper insights of these sample selection methods, we will also closely examine the examples selected by each method, such as the percentage of positive examples.

8.4 Domain Adaptation in Other Areas

We will extend the application of the proposed approaches form cross-domain video concept detection to other adaptation problems within and beyond multimedia. We identify several potential applications. One application is the adaptation of multimedia retrieval models trained from a specific corpus, query type, or query to other corpora, query types, and queries. Another application is the adaptation of classifiers for text categorization across different text corpora, in order to evaluate our approach on high-dimensional, discrete term-vector features of text documents.

8.5 Timeline

- **Nov.- Dec. 2007:** Conduct an empirical survey on the impact of domain change to video concept detection.
- **Jan. 2007 - Mar. 2008:** Implement and evaluate domain analysis approaches.
- **Apr. 2008:** Implement and evaluate disagreement based sample selection method.
- **May.2008 - Jul. 2008:** Apply the proposed approaches to other adaptation problems including multimedia retrieval model adaptation and cross-corpus text categorization.
- **Aug. 2008 - Sep. 2008:** Summarize and thesis write-up.

References

[1] A. Amir and et al. IBM research TRECVID-2003 video retrieval system. In *Worshop of TRECVID 2003*, 2003.

[2] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proc. of the 24th Int'l Conf. on Machine learning*, pages 17–24, 2007.

[3] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *Advances in Neural Information Processing Systems*, 19, 2007.

[5] M. Bacchiani and B. Roark. Unsupervised language model adaptation. In *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, 2003.

[6] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003.

[7] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3, 2002.

[8] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Proc. of Computational Learning Theory*, 2003.

[9] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proc. of the 24th Int'l Conf. on Machine Learning*, pages 81–88, 2007.

[10] S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. *Advances in neural information processing systems*, 19, 2007.

[11] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proc. of 26th Annual Int'l ACM SIGIR Conf. on Research and development in informaion retrieval*, 2003.

[12] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proc. of Int'l Conf. on Machine Learning*, 2000.

[13] R. Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, 1997.

[14] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Proc. of Neural Information Processing Systems*, 2000.

[15] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[16] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10(5):1055–1064, 1999.

[17] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399, 2006.

[18] M.-Y. Chen and A. Hauptmann. Discriminative fields for modeling semantic concepts in video. In *Eighth Conference on Large-Scale Semantic Access to Content*, 2007.

[19] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[20] I. Cox, M. Miller, S. Omohundro, and P. Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. *Proc. of the 13th Int'l Conf. on Pattern Recognition*, 3, 1996.

[21] P. Cunningham, N. Nowlan, S. Delany, and M. Haahr. A case-based approach to spam filtering that can track concept drift. In *Proc. of ICCBR Workshop on Long-lived CBR Systems*, 2003.

[22] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proc. of the 24th Int'l conference on Machine learning*, pages 193–200, 2007.

[23] S. B. David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 2007.

[24] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, 18(4–5):187–195, 2005.

[25] D. Ding and B. Zhang. Probabilistic model supported rank aggregation for the semantic concept detection in video. In *Proc. of ACM Int'l Conf. on Image and Video Retrieval*, 2007.

[26] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, pages 155–161, 1996.

[27] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.

[28] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge discovery and data mining*, pages 109–117, 2004.

[29] W. Fan. Systematic data selection to mine concept-drifting data streams. In *Proc. of 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 128–137, 2004.

[30] W. Fan and I. Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In *Proc. of SIAM Data Mining Conference 2007*, 2007.

[31] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proc. of the Ninth IEEE Int'l Conf. on Computer Vision*, page 1134, 2003.

[32] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1002–1009, 2004.

[33] M. Gales and S. Young. Robust continuous speech recognition using parallel modelcombination. *IEEE Trans. on Speech and Audio Processing*, 4(5):352–359, 1996.

[34] J. Gauvain and C. Lee. Maximum a posteriori estimation for multivariate Gaussian mixtureobservations of Markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2):291–298, 1994.

[35] D. Gildea. Corpus variation and parser performance. In *Conf. on Empirical Methods in Natural Language Processing*, pages 167–202, 2001.

[36] Z. Guo, Z. Zhang, E. P. Xing, and C. Faloutsos. Enhanced max margin learning on multimodal data mining in a multimedia database. In *Proc. of 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2007.

[37] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction.* Springer, 2001.

[38] A. Hauptmann and et al. Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In *Workshop of TRECVID*, 2003.

[39] G. Heitz, G. Elidan, and D. Koller. Transfer learning of object classes: From cartoons to photographs. In *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*, 2005.

[40] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2007.

[41] R. Hwa. Supervised grammar induction using training data with limited constituent information. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 73–79, 1999.

[42] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proc. of the 26nd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 119–126, 2003.

[43] T. Joachims. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines*. 1998.

[44] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61(1-3):151–165, 2005.

[45] L. S. Kennedy, A. P. Natsev, and S.-F. Chang. Automatic discovery of query-class-dependent models for multimodal search. In *Proc. of the 13th annual ACM Int'l Conf. on Multimedia*, pages 882–891, 2005.

[46] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *Proc. of Int'l Conf. on Machine Learning*, pages 487–494, 2000.

[47] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. In *Proc. of Int'l Conf. on Data Mining*, page 123, 2003.

[48] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proc. of Int'l Conf. on Machine Learning*, 2004.

[49] C. Leggetter and P. Woodland. Flexible Speaker Adaptation Using Maximum Likelihood Linear Regression. In *Proc. ARPA Spoken Language Technology Workshop*, pages 104–109, 1995.

[50] X. Liao, Y. Xue, and L. Carin. Logistic regression with an auxiliary data source. In *Proc. of Int'l Conf. on Machine Learning*, pages 505–512, 2005.

[51] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *Proc. of the 24th Annual Int'l ACM SIGIR conf. on Research and Development in Information Retrieval*, pages 267–275, 2001.

[52] Z. Marx, M. Rosenstein, L. Kaelbling, and T. Dietterich. Transfer learning with an ensemble of background tasks. In *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*, 2005.

[53] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *Proc. of the 15th Int'l Conf. on Machine Learning*, pages 350–358, 1998.

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

[55] M. R. Naphade, T. Kristjansson, B. Frey, and T. Huang. Probabilistic multimedia objects Multijects: A novel approach to video indexing and retrieval in multimedia systems. In *Proc. of ICIP*, 1998.

[56] A. P. Natsev, M. R. Naphade, and J. R. Smith. Semantic representation: search and mining of multimedia content. In *Proc. of the tenth ACM SIGKDD Int'l conference on Knowledge discovery and data mining*, pages 641–646, 2004.

[57] A. P. Natsev, M. R. Naphade, and J. Tesic. Learning the semantics of multimedia queries and concepts from a small number of examples. In *Proc. of 13th Annual ACM Int'l Conf. on Multimedia*, pages 598–607, 2005.

[58] A. Niculescu-Mizil and R. Caruana. Learning the Structure of Related Tasks. In *Proc. of NIPS-2005 Workshop on Inductive Transfer*, volume 10, 2005.

[59] J. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, 1999.

[60] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. In *Proc. of the 15th ACM Int'l Conf. on Multimedia*, pages 17–26, 2007.

[61] G. Quenot and S. Ayache. TRECVID 2007 collaborative annotation. In *http://mrim.imag.fr/tvca/*.

[62] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proc. of the Twenty-Fourth Int'l Conf. on Machine Learning*, 2007.

[63] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proc. of Int'l Conf. on Machine Learning*, pages 713–720, 2006.

[64] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2005.

[65] B. Roark and M. Bacchiani. Supervised and unsupervised pcfg adaptation to novel domains. In *Proc. of the 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 126–133, 2003.

[66] M. Rosenstein, Z. Marx, L. Kaelbling, and T. Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*, 2005.

[67] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-basedimage retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[68] S. Satoh and T. Kanade. Name-it: Association of face and name in video. In *Proc. of the Conf. on Computer Vision and Pattern Recognition*, pages 368–373. IEEE Computer Society, 1997.

[69] R. E. Schapire, M. Rochery, M. G. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *Proc. of the 9th Int'l Conf. on Machine Learning*, pages 538–545, 2002.

[70] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. of the 7th Int'l Conf. on Machine Learning*, pages 839–846, 2000.

[71] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proc. of the fifth Aannual Workshop on Computational learning theory*, pages 287–294, 1992.

[72] D. Shen, J. Zhang, G. Zhou, J. Su, and C.-L. Tan. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proc. of the ACL 2003 workshop on Natural language processing in biomedicine*, pages 49–56, 2003.

[73] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[74] A. Smeaton and P. Over. Trecvid: Benchmarking the effectiveness of infomration retrieval tasks on digital video. In *Proc. of Conf. on Image and Video Retrieval*, 2003.

[75] C. Snoek, M.Worring, J. Geusebroek, D. Koelma, and F. Seinstra. The mediamill trecvid 2004 semantic viedo search engine. In *Proc. of TRECVID*, 2004.

[76] W. N. Street and Y. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proc. of ACM SIGKDD Int'l Conf. on Knowledge discovery and data mining*, pages 377–382, 2001.

[77] M. Sugiyama and K. Muller. Model selection under covariate shift. Springer, 2005.

[78] N. Syed, H. Liu, and K. Sung. Incremental learning with support vector machines. In *In Workshop on Support Vector Machines, at the IJCAI*, 1999.

[79] M. Szummer and R. Picard. Indoor-outdoor image classification. In *IEEE Int'l Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV*, volume 98, pages 42–51, 1998.

[80] M. Taylor and P. Stone. Cross-domain transfer for reinforcement learning. *Proc. of the 24th Int'l conference on Machine learning*, pages 879–886, 2007.

[81] S. Thrun. Is learning the *n*-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems*, volume 8, pages 640–646, 1996.

[82] S. Tong and D. Koller. Active learning for parameter estimation in bayesian networks. *Advances in Neural Information Processing Systems*, 13:647–653, 2000.

[83] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proc. of Int'l Conf. on Machine Learning*, pages 999–1006, 2000.

[84] L. Uebel and P. Woodland. An Investigation into Vocal Tract Length Normalisation. In *Proc. of Eurospeech-99*, 1999.

[85] A. Vailaya, M. Figueiredo, and A. Jain. Image classification for content-based indexing. *IEEE Trans. on Image Processing*, 10(1):117–130, 2001.

[86] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. of ACM SIGKDD Int'l Conf. on Knowledge discovery and data mining*, pages 226–235, 2003.

[87] T. Westerveld, T. Ianeva, L. Boldareva, A. de Vries, and D. Hiemstra. Combining information sources for video retrieval. In *TRECVID 2003 Workshop*, 2004.

[88] P. Wu and T. G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *Proc. of Int'l Conf. on Machine Learning*, page 110, 2004.

[89] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proc. of the 12th annual ACM Int'l Conf. on Multimedia*, pages 572–579, 2004.

[90] E. Xing, R. Yan, and A. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proc. of the 21th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-05)*, 2005.

[91] B. L. T. Y. Wu and J. R. Smith. Ontology-based multiclassification learning for video concept detection. In *IEEE Int'l Conference on Multimedia and Expo*, 2004.

[92] R. Yan. Probabilistic models for combining diverse knowledge sources in multimedia retrieval. In *Ph.D Thesis*, 2006.

[93] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *Proc. of IEEE Int'l Conf. on Computer Vision*, pages 516–523, 2003.

[94] R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proc. of the 12th ACM Int'l Conf. on Multimedia*, pages 548–555, 2004.

[95] R. Yan, M. yu Chen, and A. G. Hauptmann. Mining relationship between video concepts using probabilistic graphical model. In *IEEE Int'l Conf. on Multimedia and Expo*, 2006.

[96] J. Yang and A. Hauptmann. Annotating news video with locations. In *Proc. of 5rd Int'l Conf. on Image and Video Retrieval*, 2006.

[97] J. Yang and A. G. Hauptmann. Naming every individual in news video monologues. In *Proc. of the 12th annual ACM Int'l Conf. on Multimedia*, pages 580–587, 2004.

[98] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *Proc. of the 22nd Int'l conference on Machine learning*, pages 1012–1019, 2005.

[99] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proc. of the 21st Int'l Conf. on Machine learning*, page 114, 2004.

[100] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems 18*, pages 1585–1592, 2006.

[101] Y. Zhang. Using bayesian priors to combine classifiers for adaptive filtering. In *Proc. of the 27th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 345–352, 2004.