Image Classification with Minimal Supervision

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Ajay Jayant Joshi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nikolaos P. Papanikolopoulos, Adviser

June 2011

Acknowledgments

The work in this thesis has been generously supported by the NSF, MnDOT, ITS, and the graduate school through the Doctoral Dissertation Fellowship program.

I am grateful to many people for helping me get to this point. Many thanks to Nikos for his support, encouragement, and guidance over the last 5 years. Apart from being my thesis adviser, he has also been a great mentor and advocate, and I have learned a lot from him. I am extremely grateful for the immense freedom he provided in all aspects of my graduate school experience. I had a lot of fun in school, and much of the credit goes to him.

Thanks to Guillermo, Arindam, and Maria for serving on my thesis committee, and for their advice at various points during the work. I am especially grateful to Arindam for introducing me to the fascinating field of Machine Learning, which I have come to love. It was always a pleasure attending his classes and having technical discussions with him. Arindam's excitement about Machine Learning is truly infectious and has inspired me throughout.

Thanks to Osama Masoud who guided me through my first semester in the lab, and Vassilios Morellas for being a good sounding board for new ideas. A big thanks to all my lab-mates and other folks at school; all of you made graduate life a fun and stimulating experience. I made many friendships in all these years, and I will cherish them for the rest of my life.

Thanks to Georganne for answering all sorts of questions about the program and preparing my DDF nomination, and to Naila for processing endless travel requests and reimbursements. I would also like to thank folks at the Digital Technology Center for their support throughout.

Life away from home was only possible due to the love and support of all my friends, right from school to a bunch of wonderful people here in Minneapolis. Words can't capture the immaculate support system they provided, be it on the phone halfway across the globe or over a cup of coffee nearby. Somehow, they always seemed to be around when I needed them most. Thanks to many friends at the Continental Cricket Club who made me feel at home when I first got to the Twin Cities.

I am indebted to Dk for convincing me to pursue a PhD in the first place. Dk's

appetite for research and new ideas continues to amaze me to this day. Discussions with him at Starbucks were truly inspiring, both technically and otherwise. After all these years, I am now convinced that the choice was right!

An internship at MERL was my first experience outside school, and Dr. Fatih Porikli made it exceptional. I am extremely grateful for his support and advice ever since.

I will always cherish my time in Minneapolis; it has given me many fond memories – late nights at Lake Harriet and the Uncommon Grounds Cafe, evenings in coffee shops all around town, never-ending board games, squash evenings, bike rides, snowstorms, the parks, restaurants, lakes, and of course all the great people. Thanks to everyone for making the last few years extremely enjoyable and memorable.

Finally, everything I have today is due to the unwavering support of Aai, Baba, and Gayatri. I will be forever indebted to them for the unconditional love, friendship, and care they have provided through all my endeavors.

Abstract

With growing collections of images and video, it is imperative to have automated techniques for extracting information from visual data. A primary task that lies at the heart of information extraction is image classification, which refers to classifying images or parts of them as belonging to certain categories. Accurate and reliable image classification has diverse applications – web image and video search, content based image retrieval, medical image analysis, autonomous robotics, gesture-based human computer interfaces, etc.

However, considering the large image variability and typically high-dimensional representations, training predictive models requires substantial amounts of annotated data, often provided through human supervision – supplying such data is expensive and tedious. This training bottleneck is the motivation for development of robust algorithms that can build powerful predictive models with little training or supervision.

In this thesis, we propose new algorithms for learning with data, particularly focusing on active learning. Instead of passively accepting training data, the basic idea in active learning is to select the most informative data samples for the human to annotate. This can lead to extremely efficient allocation of resources, and results in predictive models that require far fewer training samples compared to the passive setting.

We first propose an active sample selection criterion for training large multi-class classifiers with hundreds of categories. The criterion is easy to compute, and extends traditional two-class active learning to the multi-class setting.

We then generalize the approach to handle only binary (yes / no) type feedback while still performing classification in the multi-class domain. The proposed modality provides substantial interactive simplicity, and makes it easy to distribute the training process across many users.

Active learning has been studied from two different perspectives: selective sampling from a pool, and query synthesis; both perspectives offer different tradeoffs. We propose a formulation that combines both approaches while leveraging their individual strengths resulting in a scalable and efficient multi-class active learning scheme.

Experimental results show efficient training of classification systems with a pool of a few million images on a single computer.

Active learning is intimately related to a large body of previous work on experiment design and optimal sensing – we discuss the similarities and key differences between the two. A new greedy batch-mode sample selection algorithm is proposed that shows substantial benefits over random batch selection, when iterative querying cannot be applied.

We finally discuss two applications of active selection: i) active learning of compact hash codes for fast image search and classification, and ii) incremental learning of a classifier in a resource-constrained environment to handle changing scene conditions.

Throughout the thesis, we focus on thorough experimental validation on a variety of image datasets to analyze strengths and weaknesses of the proposed methods.

Contents

List of Tables

List of Figures

1 Introduction

Visual data consisting of images and video are everywhere around us – on the world wide web, in the medical imaging domain, in personal collections, from visual surveillance cameras, in the motion picture industry, etc. Research in the field of Computer Vision aims to develop computer systems that automatically extract semantic meaning from image and video pixels. A number of important applications depend on the success of computer vision techniques – e.g., computerized analysis of brain CT scans for detection of malicious tumors, detection of suspicious activity in video, interfaces for the visually impaired, web image search, gesture recognition for human-machine interaction, autonomous driving, robotic rescue operations, etc.

It is believed that images and video will occupy the majority of all digital storage, including the world wide web, within the next few years. Even today, the numbers are staggering: according to Facebook [2010], *every month* more than *3 billion* photos were uploaded on their servers even a year ago. Surveillance video and medical image data fill up gigabytes every day. It is clear that manual analysis cannot handle such astounding scales of data, and it is imperative to study computational techniques for automated information extraction.

We focus on a sub-problem of automated analysis of visual content – image classification. The goal is to devise a system that can classify images or image parts into categories based on the content, e.g., types of objects, outdoor scenes, people, handwriting, etc. The task itself may vary depending on the application of interest and can be extended to include video data as well. For instance, the task might be to determine the outdoor or indoor scene that each of the following images shown in Figure 1 belongs to.

Typically, machine learning techniques are used to learn statistical models based on *annotated training data*, and the model can then predict the categories of data samples seen in the future. The main goal of this thesis is to devise machine learning

Building Highway Kitchen Office Mountain

Figure 1: Each image is to be classified to determine whether it belongs to a set of defined categories.

algorithms that can learn predictive models with little training data.

1.1 Organization of the thesis

The thesis is organized as follows. In Chapter 2, we discuss the challenges making image classification and content analysis a hard problem. The main objectives of the thesis are also outlined in the chapter. In Chapter 3, we give a detailed overview of the related work on various aspects including image features and representation, object recognition and other applications, semi-supervised leaning and active learning approaches. We also briefly discuss other relevant work in experiment design.

Chapter 4 proposes a simple active selection measure that can be applied to multi-class problems, and outperforms conventional methods such as entropy-based selection. In order to simplify interaction for providing user annotation, in Chapter 5, we explore a *binary yes/no feedback modality for multi-class* classification that improves interactive efficiency of the training process.

In Chapter 6, we propose scalable approximate active learning methods for efficiently handling very large datasets. There have been two broad approaches to active learning: membership query synthesis, and selective sampling from a pool. Chapter 7 proposes a new formulation that combines both the approaches by exploiting the advantages of both. In particular, the speed of query synthesis is retained, while at the same time, *meaningful* samples are chosen from the pool.

Chapter 8 outlines relations between active learning and experiment design, and discusses similar work in the field of optimal sensing. We also propose greedy, batch-mode active selection algorithms that outperform random selection, and are applicable

where iterative learning is otherwise not possible. Finally, the thesis explores applications of active selection in sequentially learning compact hash codes in Chapter 9 and in incremental classifier design for changing scene conditions in resource-constrained environments in Chapter 10. Chapter 11 concludes the thesis and discusses potential directions for future work.

2 Challenges in Image Classification

With the huge increase in visual data in the form of images and video, our reliance on automatic analysis is continuously rising. One primary problem in analyzing visual data is the task of image classification, i.e., classifying an image or a part of it into a certain category. Classifying images is challenging since it is difficult to capture the semantic content of images with numerical representations. Further, achieving performance similar to humans is hard as human understanding of visual data is highly context driven and works at multiple levels in the semantic hierarchy. Because of the wide variety of image appearances and their content, it is virtually impossible to encode the image classification problem as a set of rules. Thus, there has been a lot of interest in using machine learning techniques to address the classification task.

Learning techniques usually require large quantities of human annotated training data for satisfactory performance. Images representing the same content are often taken in very different scene conditions – varying illumination, camera angle, pose of the subject/object, apparent size and color, etc. The significant variation in scene conditions can hamper the generalization ability of learning techniques. In order to counter for the variation, training data representing a large range of scene conditions is required. Annotating (or labeling) images is extremely time consuming and very repetitive for humans. Further, in many cases, enough training data might simply not be available for annotation. For example, consider the problem of detecting suspicious activity in video. Most recorded video data consists of normal activity, and enough examples of suspicious activity are not available to train the system. It is therefore important to be able to perform well with little training data.

2.1 Image representations

One of the most challenging aspects of image and video classification is choosing the right representation. Unlike documents wherein 'words' form informative units of rep-

resentation that can be used for highly accurate classification, it is not easy to define 'units' of representation for images. For example, pixels often do not capture context or neighborhood structure that is extremely important for recognition (although in some easier tasks such as digit recognition, pixel values are typically used). Image patches have also been explored since they capture some spatial context and possibly informative statistics local to that region of the image. However, choosing the appropriate patch sizes is dependent on the image content, which may not be known beforehand. Furthermore, scale variations and camera positions often distort image patches to make comparisons harder. Even if patches form appropriate descriptors for a certain domain, other typical variations like changes in illumination conditions, focus (out of focus), motion blur, etc. can adversely affect the classification quality offered by the descriptors.

More comprehensive image descriptors such as keypoint representations, bag-of-features style descriptors have been studied which typically require substantially more computation. However, this is still a very active research area and forms one of the most challenging aspects of recognizing and classifying visual data.

2.2 Capturing context

As mentioned before, context is an important cue in classification that feature representations often fail to capture. For example, color and edge gradient descriptors cannot differentiate between images of sky versus sea. As such, it is extremely important to incorporate other contextual information which can be used for such discrimination. In news articles for instance, text and images occur simultaneously and often refer to similar topics or themes. Such co-occurrences can be used for context-based classification and recognition.

2.3 The need for statistical learning

Due to the complex feature representations (high dimensional vectors, sets, bags of features, etc.) it is hard to define precise rules for classification. The problem is even more acute in the presence of typical variations of lightning, and noise in the data. As a consequence, most of the successful results in image classification and recognition

involve explicitly training statistical models from annotated samples. Given enough data, these models can capture highly complex relationships between feature values and their correlations with image content. With the help of features that are relatively invariant to geometric and illumination variations, the statistical machine learning approaches provide classification accuracies much better than hand-trained models.

On the other hand, the need for large amounts of annotated data is usually the bottleneck in training statistical models. Even though image data is abundant, annotation requires human effort and supervision, which is expensive and time-consuming. The motivation for the work in this thesis comes from this bottleneck of providing user supervision. The main idea is to develop learning systems that learn from interactions with the environment formalized through a querying mechanism. By having humans in the training loop, interaction provides extremely efficient ways of learning models that outperform passive methods.

2.4 Performance requirements

Most applications of visual classification have stringent performance requirements, both in terms of time and human effort. Due to the large cost associated with annotating samples, crowd-sourcing systems such as Amazon's Mechanical Turk [mtu] are used to obtain (relatively noisy) annotations, obtained in exchange for a small charge per annotation. Thus, there are real monetary costs involved in training. Aside from the training, during "test time" the system needs to respond quickly for practical use cases. For example, a robotic vision system used for navigation needs to be fast enough to allow real-time navigational decision-making. A search engine is expected to have millisecond response times for typical web-search applications.

Since most applications (especially, web-related ones) involve very large quantities of data, the systems need to be scalable to efficiently handle huge data sizes, with low time and memory requirements.

To summarize, image classification with minimal training or supervision poses many interesting challenges. Some of the challenges that make visual classification a particularly hard problem are summarized below.

- It can be difficult to extract image representations that capture discriminat-

ing information between different image categories, particularly, since human judgment often forms the baseline for defining what a category consists of.

- Image and video properties have significant dependence on scene conditions. For example, the same object might look very different under different conditions of illumination, camera properties, viewing angle, background, etc. Hence, a classifier trained to recognize visual categories in one environment might perform poorly when deployed in another.

- Because of the diversity in scene conditions and high-dimensional image representations, huge training sets are required to achieve satisfactory performance.

- Many real-world image classification problems consist of a large number of categories. Consider for example an object category recognition system that continuously encounters new object categories with time. It is imperative to devise learning algorithms that can handle many categories or classes.

- Many computational bottlenecks exist due to the number of images available and large image sizes. In order for the learning algorithms to be practical, efficient classification methods are essential.

One the other hand, there are some aspects of visual data that can be exploited. Even though annotated visual data is scarce, a huge amount of unannotated data is often available, especially on the web. A large benefit can be obtained if this data is effectively utilized. Further, images are easy to visualize for humans, and annotation is often possible in a real-time interface. Also, annotating images is easier than tuning system parameters for many applications.

Based on the above observations, our objective is to develop active learning algorithms for visual classification that can optimally exploit whatever little human input that is available. We summarize our objectives below. Note that we primarily focus on the *classification* setting in this thesis since most of the applications explored here adhere to classification – some analogous work has been done in the regression setting, which we mention where applicable.

2.5 Our objectives

Our broad objectives in this thesis are to propose new active learning algorithms that reduce the training time and effort required to train large-scale image classification systems. Some of the primary considerations are:

- The algorithms need to be scalable to very large amounts of data, and be fast enough so as to allow real-time human interaction.
- Modalities for providing annotations need to be simple and easy to distribute, so that annotations can be done across a large number of users simultaneously.
- Since real-world classification tasks typically have a large number of categories, the methods need to address multi-class problems. Traditionally, a fairly restrictive assumption has been made while evaluating multi-class classification; specifically, it is assumed that the system has access to annotated samples from *all categories of interest to begin with.* In typical applications, enforcing this restriction is often impossible. We do not make any such assumption in our work so that the system can encounter newer categories as it explores the space.
- The system needs to be robust to issues common in real data such as noise (or labeling errors), class population imbalance, wherein the proportion of samples belonging to each class can widely vary, etc.
- We focus on thorough experimentation of the proposed ideas in realistic settings on a wide variety of image datasets including letter and digit recognition, object recognition, pedestrian detection, outdoor scene recognition, etc. Note also that most of the developed methods are relatively agnostic to the kind of data available, and some of our experiments also report results on non-image data.

We do not address the multi-label[1] classification problem in this thesis. The primary reason is that in most image classification applications, classification is performed separately for small image windows in which only one object is expected to be present. Image window descriptors are very common as they perform better

[1]Classification problems that admit multiple labels to each example are referred to as multi-label problems.

than generic global image descriptors, except for certain applications such as scene recognition in images, for which we use global image descriptors as will be discussed later.

The various algorithms presented in the thesis target different settings and requirements commonly occurring in applications. The results demonstrate the promise of the general idea of active learning in the image classification domain for a wide variety of tasks.

3 Related Work

In this section, we provide a comprehensive review of the literature on image classification including image feature descriptors and various machine learning approaches such as semi-supervised and active learning for various application like object recognition, human detection, handwriting recognition, etc. The categorization below is done primarily for ease of understanding, and many techniques in the literature span multiple categories.

3.1 Object detection and recognition

The objective of object recognition is to classify images based on the object(s) they contain. We differentiate between recognition and detection in the sense that detection aims to *locate* the objects of interest in images, whereas recognition aims to *classify* images based on the objects of interest. Thus, holistic image features can be used for recognition, however, detection requires precise models for *locating* the objects. There has been significant research over the past decade on extracting useful features and also on employing learning techniques for object detection and recognition.

3.1.1 Interest point detection and representations

Finding points of interest in images and coming up with succinct feature representations is a crucial part of accurate object recognition. Achieving invariance to scale, lighting, pose, and other such factors while maintaining the representative power of features has been the primary theme of research on image representations. A popular method used heavily in practice is the Scale Invariant Feature Transform (SIFT) proposed by Lowe [2004]. The basic idea is to find scale space extrema by comparing smoothed images at different scales. As a result invariant features are obtained

that can be used for finding correspondences across two images with significant viewpoint variation. We use the SIFT representation for some of our object recognition experiments.

Gradient values of pixel intensity in an image are a vital cue for identifying image content. At a certain resolution, different objects lead to different gradient distributions, and they can thus be used for discriminating between object classes. A popular gradient-based representation, particularly applicable for detecting humans in still images is the Histogram of Oriented Gradients (HoG) proposed by Dalal and Triggs [2005]. The HoG representation has been extended to pyramid models called Pyramid Histogram of Oriented Gradients (PHOG) by Bosch et al. [2007]. We employ the HoG representation for our experiments on human detection in videos, and the PHOG features for object recognition.

Berg and Malik [2001] propose Geometric Blur (GB) features for template matching across images with relative viewpoint and shape invariance. The basic idea is to blur the image signal with spatially varying blur kernels at points of interest. The descriptor obtained can be used to find point correspondences across images. Also, image similarity can be computed with the extracted features by using an appropriate kernel function such as the one proposed in [Grauman and Darrell, 2005]. We demonstrate some experiments with the GB features for object recognition tasks.

Many other detectors and descriptors have been proposed in the literature such as Kadir-Brady saliency detector [Kadir and Brady, 2001], shape contexts [Belongie et al., 2002], spatial pyramids [Lazebnik et al., 2006], etc. For a comprehensive review on interest point detectors, see [Mikolajczyk and Schmid, 2004, and references therein].

3.1.2 Learning in object recognition

Learning methods for object category recognition have received significant attention in the past decade. Standard datasets are often used for evaluating the performance of object recognition methods. Some popular datasets are the Caltech-101 (Caltech-256) consisting of 101 (256) object categories, Oxford building dataset, MIT LabelMe, the Pascal dataset, USPS, Pendigits, and Letter datasets from the UCI repository [Asuncion and Newman, 2007].

Weber et al. [2000b] propose a method to learn object shape models assuming that objects are represented as flexible arrangements of rigid parts. Within-class variation is modeled using a joint probability distribution over the arrangement of object parts and the output of part detectors. After identifying distinctive parts in the training set, shape models are obtained through expectation-maximization. As a follow up to the previous method, Weber et al. [2000a] proposed an approach for learning of object categories based on joint probability distribution over shape models of object parts and their appearances. The part models provide some degree of robustness to image clutter. For learning the parameters of the generative model, a large set of training images was obtained through synthetic blending of the objects of interest (human head, in this case) with different backgrounds. Through synthetic blending, a large range of scene variations and clutter were introduced in the training set.

Scale-invariant models for object class recognition have been proposed by Fergus et al. [2003]. A probabilistic representation was used was object shape, appearance, occlusion and relative scale. Based on a training set of a certain category, the method finds parameters of the part-based object model. Improving on this, Fei-Fei et al. [2006] demonstrated a Bayesian scheme for learning object categories with very few training examples per category. The basic idea is that prior information obtained about previously learned categories can be used to form suitable prior distributions, which can then be modified with few training images.

Zhang et al. [2006] perform visual category recognition through SVM-KNN, a combination of Support Vector Machines (SVM [Vapnik, 1998]) and the k-nearest neighbor (k-NN) algorithm. Specifically, for any given test image, a set of k nearest neighbors from the training set is obtained and a SVM is trained on this set. Classification of the test image is the output of the trained SVM. Experiments suggest that such a combination of SVM and k-NN can perform more accurate classification as compared to both SVM and k-NN individually, and needs fewer training examples to perform comparably to SVM. Although the classification performance of SVM-KNN is much better, the time required to classify a test image is more than that of SVM since it includes finding nearest neighbors from the entire training set.

Some methods have also been proposed for completely unsupervised object category learning. For example, probabilistic Latent Semantic Analysis (pLSA) and

Latent Dirichlet Allocation (LDA) models have been used for unsupervised object category learning by Sivic et al. [2005]. Visual words are formed by vector quantizing SIFT-like descriptors for image regions. The method shows promising results on a few image classes with some clutter. Another approach that uses pLSA and extends it to incorporate spatial information has been proposed by Fergus et al. [2005]. Their approach utilizes the results of text-based image search for object category learning, and therefore is not completely unsupervised. The principal challenge is to achieve robustness to unrelated images returned by text-based image search engines. Grauman and Darrell [2006] propose a method for unsupervised clustering of images using unordered sets of local features by computing pairwise image affinities and spectral clustering. Interestingly, the method can also incorporate human input if available.

Even though some of the methods discussed are "unsupervised" in terms of learning model parameters, they still require particular training sets for each category that needs a lot of human annotation. Notably, the methods in [Sivic et al., 2005; Grauman and Darrell, 2006] are completely unsupervised, and no human input is required. From previous studies, it is generally believed in the community that completely autonomous object category recognition is currently out of reach in realistic scenarios with a large number of image categories and significant background clutter. One of our motivations for using semi-supervised and active learning paradigms stems from this belief about the limitations of current unsupervised image classification approaches.

3.2 Scene classification

This section describes work that deals with image classification by using either holistic representations or other methods to incorporate contextual information about the image. Torralba [2003] demonstrates the importance of context in visual recognition tasks; humans use global image features along with local features even for object identification. In order to incorporate cues about context, Torralba et al. [2006] proposed a contextual model for object detection using boosted random fields. Visual scenes have also been described through Dirichlet process models [Sudderth et al., 2006], graphical models of features and parts [Murphy et al., 2003], and hierarchical

models of objects and parts [Sudderth et al., 2008].

It has been observed that humans can quickly identify natural scene categories, irrespective of the image complexity. In order for an automatic mechanism to quickly identify scene categories, it is important to classify images based on their holistic attributes only. Oliva and Torralba [2006] propose a global feature representation of a scene, called GIST, in order to classify natural scene categories. The idea is to provide a statistical summary of spatial layout properties of the image, without using segmentation or any form of grouping. Low dimensional representation of the scene called the Spatial Envelope [Oliva and Torralba, 2001] is used to find the GIST features. Many approaches using GIST features show good classification performance on a number of natural scene categories when enough training data is provided. We use the GIST features in our experiments on scene classification.

An interesting approach for multi-label active learning for scene classification has been demonstrated by Qi et al. [2008]. They employ active selection along two dimensions – the examples and the labels. Thus, example-label pairs are selected for query instead of only examples, and correlations between different classes are exploited. A multi-label Bayesian classification error bound is then minimized. Their results show that label correlation can be exploited for active learning in multi-label problems.

A research area of significant interest is content-based image retrieval (CBIR). A query on a typical text-based image search engine returns a number of unrelated images, since the search is done based on only the image captions. CBIR techniques attempt to find relevant images to a query by using image content rather than text captions. Active learning has been explored for image retrieval by some researchers [Hoi et al., 2008; Jing et al., 2004; Tong and Chang, 2001; Tong and Koller, 2001; Zhou et al., 2004]. The basic idea is to minimize user feedback on image search results to quickly return relevant images.

Tong and Chang [2001] use SVM in a relevance feedback framework for image retrieval. Their method uses margins of unlabeled examples as an uncertainty indicator for active image selection. Only binary (two-class) classification is considered. Tong and Koller [2001] propose an active selection scheme that minimizes the version space[2]

[2]Version space is the subset consisting of all hypotheses that are consistent with the training data [Mitchell, 1997].

at each iteration of user feedback. However, the version space method also targets binary classification only. One of our objectives is to devise active learning algorithms that can easily accommodate a large number of categories for broad applicability.

An interesting approach for image retrieval using batch-mode active learning has been introduced by Hoi et al. [2008]. Batch-mode active learning deals with the selection of multiple examples at each iteration. Some crucial points need to be considered for selecting images in batches, as opposed to selecting one image at each iteration. Note that images can be redundant in terms of how they impact a classifier. Therefore, two images having the highest scores with some notion of informativeness might not be very informative jointly, due to their redundancy. For batch-mode selection of m images, it is important to find the best *set* of m images rather than choosing the m best images according to their individual scores. Hoi et al. [2008] employ a measure of informativeness for a set of images and then use approximation techniques for actively selecting the best set of images. However, the primary application being image retrieval, the method proposed is only suitable for binary classification.

3.3 Using unlabeled data

In many circumstances, unlabeled data is plentiful and cheap as compared to labeled data. In order to exploit the abundance of unlabeled data, a large research effort has focused on semi-supervised learning, especially in the machine learning community. Under certain assumptions, semi-supervised learning algorithms can utilize unlabeled data with small quantities of labeled data to give better classification[3] compared to purely supervised approaches [Chapelle et al., 2006]. In this section, we give an overview of the literature on semi-supervised techniques used for image classification. For a comprehensive literature review on semi-supervised learning, refer to the survey by Zhu [2005].

Blum and Mitchell [1998] introduced the co-training setting for semi-supervised learning where the examples have two distinct views (for example, for classifying webpages, the words on the page itself can form one view, while hyperlinks pointing to the page can form another). Co-training relies on two assumptions on the data – i)

[3]We focus on the classification problem only.

the two views are conditionally independent given the classification label, and ii) the two views are compatible, i.e., their predictions on the unlabeled data are the same. Nigam and Ghani [2000] showed that co-training can give smaller classification error in a text classification task compared to single-view classifiers. Muslea et al. [2002] combine active learning with multi-view semi-supervised learning in order to make the method more robust to deviations from the assumptions made by co-training algorithms. In the vision domain, Levin et al. [2003] propose a co-training method to improve the performance of visual detectors on cars using unlabeled data. In previous work, we have also employed co-training for adaptively detecting moving shadows in video sequences [Joshi and Papanikolopoulos, 2008].

Another popular approach for using unlabeled data is via semi-supervised learning on graphs [Blum and Chawla, 2001; Zhu et al., 2003; Belkin et al., 2004]. The basic idea is to form a similarity graph where nodes represent data points and pairwise edge weights between two nodes represent the similarity values between the corresponding points. A partitioning of the similarity graph then corresponds to classification of the data. Image classification through such graph methods has been demonstrated by Balcan et al. [2005] for classifying person images from a webcam, and by Bandos et al. [2006] for hyperspectral image classification.

A semi-supervised image classification approach using SVM has been proposed by Gomez-Chova et al. [2008] for remote sensing applications. Holub et al. [2005] introduced a semi-supervised framework for object classification by combining prior knowledge obtained from unlabeled data with discriminatively trained classifiers.

As mentioned before, image representations are crucial for good classification. In order to obtain more task-specific representations, there has been some work on learning image features for better classification [Dollár et al., 2007; Ferencz et al., 2008; Frome et al., 2007; Jain et al., 2006]. Some of the methods use unlabeled data to learn discriminative features; a transfer learning approach has also been demonstrated for learning representative image features [Raina et al., 2007]. We mention these works on learning image representations for completeness, and do not explore the topic further, since it is not centrally related to our proposed work.

3.4 Active learning

Active learning has been traditionally explored in the machine learning community particularly for text classification applications; early works include [Valiant, 1984; Angluin, 1988; Seung et al., 1992; Freund et al., 1997; Campbell et al., 2000; Tong and Koller, 2001]. As opposed to passive example selection, the basic idea in active learning is to select "informative examples" to query the user for labels. Through active example selection, user effort is focused on labeling the most informative examples for the particular classification task. Theoretical results show that under certain assumptions, active learning can achieve exponential reduction in the sample complexity compared to random selection for getting similar classification accuracy [Dasgupta et al., 2005; Dasgupta, 2005, 2006; Dasgupta and Hsu, 2008; Hanneke, 2009]. In order to remove the assumptions on the data, active learning algorithms that work with arbitrary forms of noise have been proposed and analyzed [Balcan et al., 2006; Dasgupta et al., 2008; Beygelzimer et al., 2009].

There has been some recent interest in applying active learning for object recognition. The primary research task is to find notions of "informativeness" for images, that are efficient to compute and work well in practice. Kapoor et al. [2007a] propose active learning in a Gaussian Process (GP) framework for object categorization. Using the GP probabilistic model for binary classification, informative images are obtained through uncertainty sampling, i.e., finding images on which the current classifier is most uncertain. The active selection process iteratively queries the user for labels of the informative examples. The method is computationally expensive due to the GP model that requires $\mathcal{O}(n^3)$ computations, where n is the training data size. Another limitation of this method comes from the way it is generalized to multi-class classification. When multiple classes are present, a binary SVM is trained for each class, such that images belong to the particular class have positive labels, and all other images have negative labels. For multi-class active learning, the method selects one example per classifier at each iteration. The selection procedure imposes two limitations – i) only a fixed number of examples can be selected at a time, the number being equal to the number of classes in the data, and ii) example selection is relatively balanced across all classifiers as each classifier is forced to select one example. Balanced se-

lection can often be a limitation, particularly so when class populations are highly imbalanced.

Holub et al. [2008] recently proposed an entropy-based active learning method for object recognition. The basic idea is to compute the expected reduction in entropy when an unlabeled image is added to the training set. The examples that lead to the maximum expected reduction in entropy are selected to query the user. The entropy-based approach is also extremely expensive, requiring $\mathcal{O}(n^3k^3)$ computations, with n unlabeled examples and k object categories. Notably, this approach allows selection of multiple images at each iteration, however, the method becomes computationally infeasible with even a few images selected. One of the challenges in active image selection is to come up with methods that are scalable to large datasets and multiple categories.

Another multi-class active learning method for video labeling has been proposed by Yan et al. [2003]. They first formulate the problem in a risk minimization framework, and then explore heuristic approximation techniques for selecting multiple examples at every iteration in a reasonable time. The paper demonstrates promising results for recognizing the person in a given image.

More recently, work in active learning has looked at requesting tags at different levels of granularity [Vijayanarasimhan and Grauman, 2008, 2009]. The motivation behind such approaches is that the cost of providing annotation can vary widely depending on the type of information provided – for example, providing an image tag is much simpler than an explicit clustering of the image. They take into account these costs of annotation along with informativeness measures for active selection.

Considering the large amounts of unlabeled data typically available, there has been recent focus on scaling up active learning for faster querying [Segal et al., 2006; Zhao et al., 2008; Vijayanarasimhan and Grauman, 2011; Jain et al., 2010].

The described methods demonstrate promising results for improving image classification, and we build upon these ideas in this thesis. The following Chapter describes the active learning setup we employ, and a simple multi-class active selection measure that empirically performs well on various image classification problems.

4 Multi-Class Active Learning

4.1 Pool-based learning setup

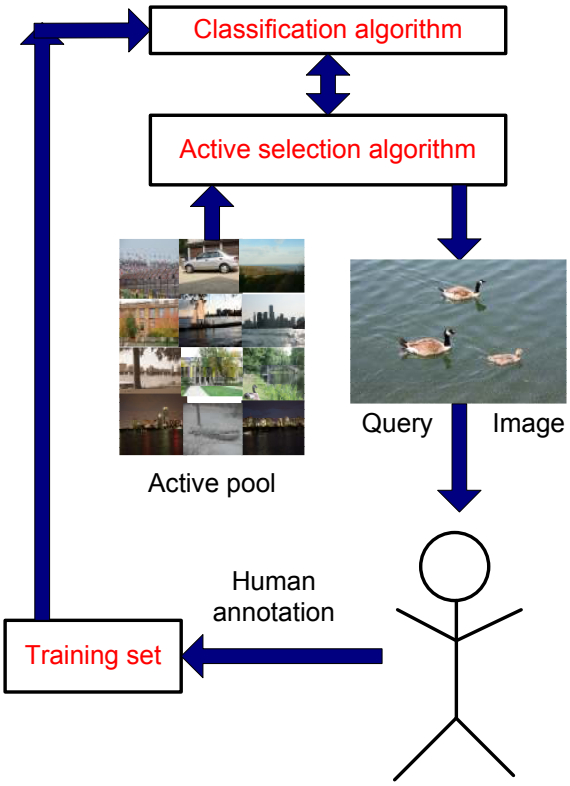


Figure 2: Flowchart of the active learning process.

Here we describe pool-based learning, which is a very common setup for active learning. We consider that a classifier is trained using a small number of randomly

selected labeled examples called the seed set. The active learning algorithm can then select examples to query the user (for labels) from a pool of unlabeled examples referred to as the active pool or the unlabeled pool. The actively selected examples along with user-provided labels are then added to the training set. This querying process is iterative such that after each iteration of user feedback, the classifier is retrained. Finally, performance evaluation is done on a separate test set different from the seed set and the active learning pool. In this Section, we use Support Vector Machines (SVM) as the primary classifier for evaluation, however, other classification techniques could potentially be employed. Figure 2[4] illustrates the overall learning setup. A similar pool-based setting is assumed throughout the thesis unless mentioned otherwise, such as in Chapter 5 which uses a slightly modified interaction method, and Chapter 8 wherein batches of samples are labeled and there is no iterative process.

Our approach follows the idea of uncertainty sampling [Campbell et al., 2000; Freund et al., 1997], wherein examples on which the current classifier is uncertain are selected to query the user. Distance from the hyperplane for margin-based classifiers has been used as a notion of uncertainty in previous work. However, this does not easily extend to multi-class classification due to the presence of multiple hyperplanes. We use a different notion of uncertainty that is easily applicable to a large number of classes. The uncertainty can be obtained from the class membership probability estimates for the unlabeled examples as output by the multi-class classifier. In the case of a probabilistic model, these values are directly available. For other classifiers such as SVM, we need to first estimate class membership probabilities of the unlabeled examples. In the following, we outline our approach for estimating the probability values for multi-class SVM. However, such an approach for estimating probabilities can be used with many other non-probabilistic classification techniques also.

4.2 Probability estimation

In order to obtain class membership probability estimates for unlabeled examples in the active pool, we follow the approach proposed by Lin et al. [2007], which is a modified version of Platt's method to extract probabilistic outputs from SVM [Platt,

[4]All figures in the thesis are best viewed in color.

2000].

The basic idea is to approximate the class probability using a sigmoid function. Suppose that $x_i \in \mathbb{R}^n$ are the feature vectors, $y_i \in \{-1, 1\}$ are their corresponding labels, and $f(x)$ is the decision function of the SVM which can be used to find the class prediction by thresholding. The conditional probability of class membership $P(y = 1|x)$ can be approximated using

$$p(y = 1|x) = \frac{1}{1 + \exp(Af(x) + B)}, \tag{1}$$

where A and B are parameters to be estimated. Maximum likelihood estimation is used to solve for the parameters:

$$\min_{(A,B)} \quad -\sum_{i=i}^{l} (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)), \tag{2}$$

where,

$$p_i = \frac{1}{1 + \exp(Af(x_i) + B)},$$

$$t_i = \begin{cases} \frac{N_p+1}{N_p+2} & \text{if } y_i = 1 \ ; \\ \frac{1}{N_n+2} & \text{if } y_i = -1. \end{cases}$$

N_p and N_n are the number of examples belonging to the positive and the negative class respectively in the training set. Newton's method with backtracking line search can be used to solve the above optimization problem to obtain the probability estimates [Lin et al., 2007].

The primary SVM classifier considered above is binary. We use the one-versus-one approach (a classifier trained for each pair of classes) for multi-class classification. The one-versus-one method for SVM is computationally efficient and shows good classification performance [Hsu and Lin, 2002]. Probability estimates for the multi-class case can be obtained through a method such as pairwise coupling [Wu et al., 2004]. In order to estimate these probabilities, we first need binary probability estimates which can be obtained from the method described above. Assume that r_{ij} are the binary probability estimates of $P(y = i|y = i \text{ or } j, \mathbf{x})$, obtained from the method above. In the multi-class case, denote the probability estimate for class i to be p_i.

Using pairwise coupling the problem can be formulated as

$$\min_{\mathbf{p}} \frac{1}{2} \sum_{i=1}^{k} \sum_{j:j \neq i} (r_{ji} p_i - r_{ij} p_j)^2, \tag{3}$$
$$\text{subject to } \sum_{i=1}^{k} p_i = 1, p_i \geq 0, \forall i,$$

where k denotes the number of classes. The above optimization problem can be shown to be convex and thereby admits a unique global minimum. It can be solved using a direct method such as Gaussian elimination, or a simple iterative algorithm. We use the toolbox LIBSVM [Chang and Lin, 2001] that implements the methods described above for classification and probability estimation in the multi-class problem.

4.3 Entropy (EP) as uncertainty

Each labeled training example belongs to a certain class denoted by $y \in \{1, \ldots, k\}$. However, we do not know true class labels for examples in the active pool. For each unlabeled example, we can consider the class membership variable to be a random variable denoted by Y. We have a distribution $\mathbf{p}$ for Y of estimated class membership probabilities computed in the way described above. Entropy is a measure of uncertainty of a random variable. Since we are looking for measures that indicate uncertainty in class membership Y, its discrete entropy is a natural choice. The discrete entropy of Y can be estimated by

$$\mathcal{H}(Y) = -\sum_{i=1}^{k} p_i \log(p_i). \tag{4}$$

Higher values of entropy imply more uncertainty in the distribution; this can be used as an indicator of uncertainty of an example. If an example has a distribution with high entropy, the classifier is uncertain about its class membership.

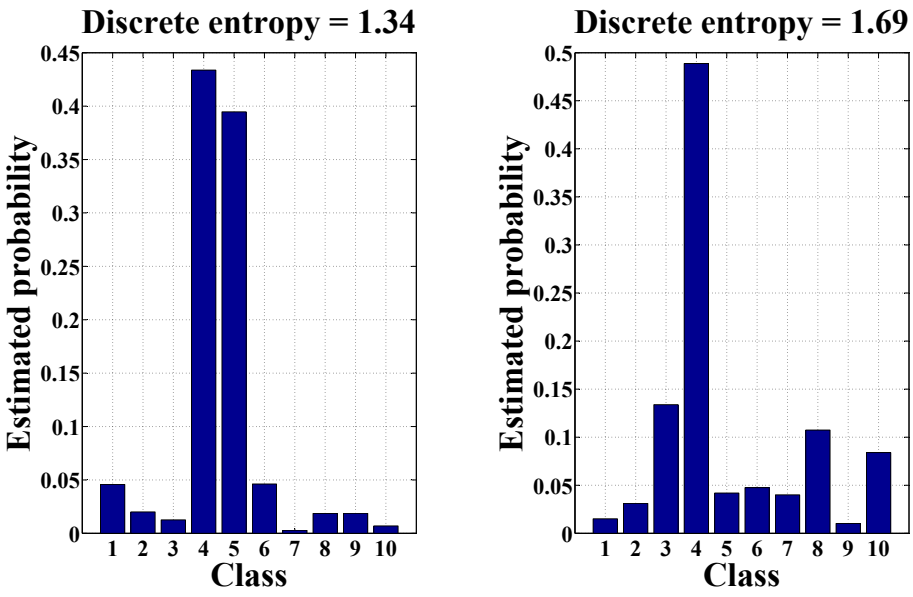
The algorithm proceeds in the following way. At each round of active learning, we

compute class membership probabilities for all examples in the active pool. Examples with the highest estimated value of discrete entropy are selected to query the user. User labels are obtained and the corresponding examples are incorporated in the training set and the classifier is retrained. As will be seen in Section 4.9, active learning through entropy (EP)-based selection outperforms random selection in some cases.

4.4 Best-versus-second best (BvSB)

Even though EP-based active learning is often better than random selection, it has a drawback. A problem of the EP measure is that its value is heavily influenced by probability values of unimportant classes. See Figure 3 for a simple illustration. The figure shows estimated probability values for two examples on a 10-class problem. The example on the left has a smaller entropy than the one on the right. However, from a classification perspective, the classifier is more confused about about the former since it assigns close probability values to two classes. For the example in Figure 3(b), small probability values of unimportant classes contribute to the high entropy score, even though the classifier is much more confident about the classification of the example. This problem becomes even more acute when a large number of classes are present. Although entropy is a true indicator of uncertainty of a random variable, we are interested in a more specific type of uncertainty relating only to classification amongst the most confused classes (the example is virtually guaranteed to not belong to classes having a small probability estimate).

Instead of relying on the entropy score, we take a more greedy approach to account for the problem mentioned. We consider the difference between the probability values of the two classes having the highest estimated probability value as a measure of uncertainty. Since it is a comparison of the best guess and the second best guess, we refer to it as the Best-versus-Second-Best (BvSB) approach. Such a measure is a more direct way of estimating confusion about class membership from a classification standpoint. Using the BvSB measure, the example on the left in Figure 3 will be selected to query the user. As mentioned previously, confidence estimates are reliable in the sense that classes assigned low probabilities are very rarely the true classes



(a) (b)

Figure 3: An illustration of why entropy can be a poor estimate of classification uncertainty. The plots show estimated probability distributions for two unlabeled examples in a 10 class problem. In (a), the classifier is highly confused between classes 4 and 5. In (b), the classifier is relatively more confident that the example belongs to class 4, but is assigned higher entropy. The entropy measure is influenced by probability values of unimportant classes.

of the examples. However, this is only true if the initial training set size is large enough for good probability estimation. In our experiments, we start from as few as 2 examples for training in a 100 class problem. In such cases, initially the probability estimates are not very reliable, and random example selection gives similar results. As the number of examples in the training set grows, active learning through BvSB quickly dominates random selection by a significant margin.

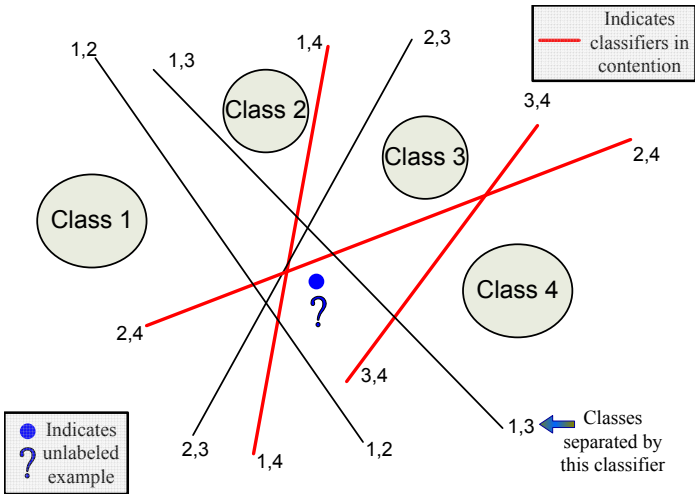


Figure 4: Illustration of one-vs-one classification (classes that each classifier separates are noted). Assuming that the estimated distribution for the unlabeled example (shown as a blue disk) peaks at 'Class 4', the set of classifiers in contention is shown as red lines. BvSB estimates the highest uncertainty in this set – uncertainty of other classifiers is irrelevant.

4.5 Another perspective

One way to see why active selection works is to consider the BvSB measure as a greedy approximation to entropy for estimating classification uncertainty. We describe another perspective that explains why selecting examples in this way is beneficial. The understanding crucially relies on our use of one-versus-one approach for multi-class classification. Suppose that we wish to estimate the value of a certain example for

active selection. Say its true class label is l (note that this is unknown when selecting the example). We wish to find whether the example is informative, i.e., if it will modify the classification boundary of any of the classifiers, once its label is known. Since its true label is l, it can only modify the boundary of the classifiers that separate class l from the other classes. We call these classifiers as those in contention, and denote them by $\mathcal{C}_l = \{C_{(l,i)} \mid i = 1, \ldots, k, i \neq l\}$, where $C_{(i,j)}$ indicates the binary classifier that separates class i from class j. Furthermore, in order to be informative at all, the selected example needs to modify the current boundary (be a good candidate for a new support vector – as indicated by its uncertainty). Therefore, one way to look at multi-class active selection for one-versus-one SVMs is the task of finding an example that is *likely to be a support vector* for one of the *classifiers in contention*, without knowing which classifiers are in contention. See Figure 4 for an illustration.

Say that our estimated probability distribution for a certain example is denoted by $\mathbf{p}$, where p_i denotes the membership probability for class i. Also suppose that the distribution $\mathbf{p}$ has a maximum value for class h. Based on current knowledge, the most likely set of classifiers in contention is $\mathcal{C}_h$. The classification confidence for the classifiers in this set is indicated by the difference in the estimated class probability values, $p_h - p_i$. This difference is an indicator of how informative the particular example is to a certain classifier. Minimizing the difference $p_h - p_i$, or equivalently, maximizing the confusion (uncertainty), we obtain the BvSB measure. This perspective shows that our intuition behind choosing the difference in the top two probability values of the estimated distribution has a valid underlying interpretation – it is a *measure of uncertainty for the most likely classifier in contention*. Also, the BvSB measure can then be considered to be an efficient approximation for selecting examples that are likely to be informative, in terms of changing classification boundaries.

4.6 Binary classification

For binary classification problems, our method reduces to selecting examples closest to the classification boundary, i.e., examples having the smallest margin. In binary problems, the BvSB measure finds the difference in class membership probability estimates between the two classes. The probabilities are estimated using Equation 1,

that relies on the function value $f(x)$ of each unlabeled example. Furthermore, the sigmoid fit is monotonic with the function value – the difference in class probability estimates is larger for examples away from the margin. Therefore, our active learning method can be considered to be a generalization of binary active learning schemes that select examples having the smallest margin.

4.7 Computational cost

There are two aspects to the cost of active selection. One is the cost of training the SVM on the training set at each iteration. Second is probability estimation on the active pool, and selecting examples with the highest BvSB score. Since we use one-vs-one SVM, we need to train $\mathcal{O}(k^2)$ classifiers for k classes. As the essence of active learning is to minimize training set sizes through intelligent example selection, it is also important to consider the cost of probability estimation and example selection on the relatively much larger active pool. The first cost comes from probability estimation in binary SVM classifiers. The estimation is efficient since it is performed using Newton's method with backtracking line search that guarantees quadratic rate of convergence. Given class probability values for binary SVMs, multi-class probability estimates can be obtained in $\mathcal{O}(k)$ time per example [Wu et al., 2004]. With n examples in the active pool, the entire BvSB computation scales as $\mathcal{O}(nk^2)$.

A typical run with seed set of 50 examples, active pool of 5000 examples, and a 10-class problem took about 22 seconds for 20 active learning rounds with 5 examples added at each round. The machine used had a 1.87 Ghz single core processor with 2 Gb of memory. All the active selection code was written in Matlab, and SVM implementation was done using LIBSVM (written in C) interfaced with Matlab. The total time includes the time taken to train the SVM, to produce binary probability values, and to estimate multi-class probability distribution for each example in the active pool at each round.

4.8 Relation to multi-class margin

Binary margin-based classifiers have an explicit notion of margin, which is proportional to the distance of a sample from the classification boundary. Due to the success

of approaches like SVM that directly seek maximum margin classification, there has been a lot of interest on extending the margin notion to multi-class classification. One such notion is that of Cramer and Singer [2001], defined in the following setting.

Let sample $x \in \mathbb{R}^n$ be a sample data point associated with the label y. The framework uses the following form for multi-class classification (notation as in [Cramer and Singer, 2001]):

$$H_M(x) = \underset{r=1}{\overset{k}{\operatorname{argmax}}}\{\bar{M}_r \cdot x\}, \tag{5}$$

where M has k rows and n columns and $\bar{M}_r$ is the r^{th} row of M. The dot product captures the classification confidence for class r. For an unlabeled sample, the classifier chooses the class that maximizes the above confidence score. Given such a classifier, the empirical error for a set of samples $S = \{(x_i, y_i), \ldots, (x_m, y_m)\}$ can then be defined in the following way:

$$\epsilon_S(M) = \frac{1}{m} \sum_{i=1}^{m} \left[H_M(x_i) \neq y_i\right]. \tag{6}$$

The goal in training is to find M that achieves a small empirical error while generalizing well to unseen data. Cramer and Singer [2001] note that direct minimization of empirical error is computationally hard, and therefore define a margin maximization approach analogous to binary SVM. The misclassification error is replaced by a piecewise linear function instead:

$$\max_r\{\bar{M}_r \cdot x + 1 - \delta_{y,r}\} - \bar{M}_y \cdot x, \tag{7}$$

where $\delta_{a,b} = 1$, iff $a = b$.

We observe from the above expression that the value of the function is zero when the classification is not only correct, but achieves a certain *margin*: i.e., the confidence values assigned to the correct label is at least one more than the confidence values assigned to any of the other classes. This loss function extends the notion of binary margin to the multi-class case.

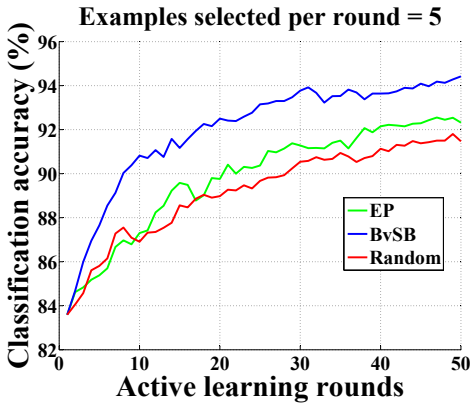
Recall that our uncertainty sampling measure is the difference in probability values

for the top two most likely classes, which is analogous to the above notion of margin, when the confidence computation is replaced by the estimated probability values. Since the above multi-class formulation extends the notion of binary margins to the multi-class case, from a similar perspective the proposed notion of BvSB extends margin minimizer uncertainty sampling in the binary setting of Campbell et al. [2000] to the multi-class setting. Furthermore, we show experiments in the following section which confirm that the proposed formulation is suitable for problems with a very large number of classes.

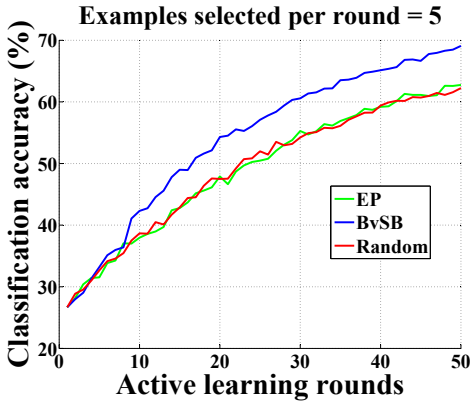
Note that even though we use the previously described notion of multi-class margin for active selection, the classifier employed is still one-vs-one SVM, since in our experiments the SVM gives better results for image classification applications of interest. Another key advantage of SVM is that it can be used with any suitable similarity function through the kernel trick. Therefore, even though the motivations are similar, the proposed active selection method is not equivalent to using the multi-class classification scheme of Cramer and Singer [2001].

4.9 Experiments with BvSB

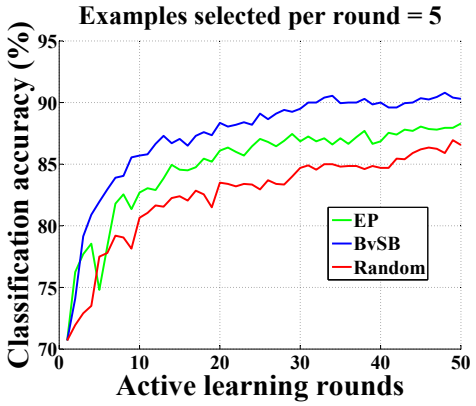
In this section, we show experiments demonstrating the ability of the BvSB measure to select informative examples for query. Later, we incorporate BvSB as an approximation to reduce the computational expense of Value-of-Information based active learning discussed in the following chapter. We demonstrate results on standard image datasets available from the UCI repository [Asuncion and Newman, 2007], the Caltech-101 dataset of object categories, and a dataset of 13 natural scene categories. All the results show significant improvement owing to active example selection. Table 1 shows a summary of datasets used and their details. For choosing the kernel, we ran supervised learning experiments with linear, polynomial, and Radial Basis Function (RBF) kernels on a randomly chosen training set, and picked the kernel that gave the best classification accuracy averaging over multiple runs.



(a)



(b)



(c)

Figure 5: Classification accuracy on (a) Pendigits, (b) Letter, and (c) USPS datasets. Note the improvement in accuracy obtained by BvSB approach over random selection. For similar accuracy, the active learning method requires far fewer training examples. In (b), EP-based selection performs poorly due to the larger number of classes.

Dataset	#classes	#features	# Pool	# Test	Kernel
USPS	10	256	5000	2000	Gaussian
Pendigits	10	16	5000	2000	Linear
Scene-13	13	320 (From [OT01]*)	5000	2000	Linear
Caltech-101	101	N/A	1515	1515	From [GD05]*

Table 1: Dataset details. # pool = active pool size, # test = test set size. [GD05]* refers to [Grauman and Darrell, 2005]. [OT01]* refers to [Oliva and Torralba, 2001].

4.9.1 Standard datasets

Figure 5(a) shows classification results on the pendigits dataset. The three methods compared are EP-based selection, BvSB-based selection, and random example selection. All three methods start with the same seed set of 100 examples. At each round of active learning, we select $n = 5$ examples to query the user for labels. BvSB selects useful examples for learning, and gradually dominates both the other approaches. Given the same size of training data, as indicated by the same point on the x-axis, BvSB gives significantly improved classification accuracy. From another perspective, for achieving the same value of classification accuracy on the test data (same point on the y-axis), our active learning method needs far fewer training examples than random selection. The result indicates that the method selects useful examples at each iteration, so that user input can be effectively utilized on the most relevant examples. Note that EP-based selection does marginally better than random. The difference can be attributed to the fact that entropy is a somewhat indicative measure of classification uncertainty. However, as pointed out in Section 4.4, the entropy value has problems of high dependence on unlikely classes. The BvSB measure performs better by greedily focusing on the confusion in class membership between the most likely classes instead.

This difference between the two active selection methods becomes more clear when we look at the results on a 26 class problem. Figure 5(b) shows classification accuracy plots on the Letter dataset, which has 26 classes. EP-based selection performs even worse on this problem due to the larger number of classes, i.e., the entropy value is skewed due to the presence of more unlikely classes. Entropy is a bad indicator

of classification uncertainty in this case, and it gives close to random performance. Even with a larger number of classes, the figure shows that BvSB-based selection outperforms random selection. After 50 rounds of active learning, the improvement in classification accuracy is about 7%, which is significant for data having 26 classes.

In Figure 5(c), we show results on the USPS dataset, a dataset consisting of handwritten digits from the US Postal Service. The performance of all methods is similar to that obtained on the Pendigits dataset shown in Figure 5(a). Active selection needs far fewer training examples compared to random selection to achieve similar accuracy.

4.9.2 Reduction in training required

BvSB selection rounds	Random selection rounds	% Reduction in # training examples
3	6	11.53
4	10	20
5	13	24.24
6	19	33.33
7	28	43.75
8	29	42.85
9	43	53.96
10	44	53.12
11	43	50.79
12	48	52.94
13	50	52.85

Table 2: Percentage reduction in the number of training examples provided to the active learning algorithm to achieve classification accuracy equal to or more than random example selection on the USPS dataset.

In this section, we perform experiments to quantify the reduction in the number of training examples required for BvSB to obtain similar classification accuracy as random example selection. For each round of active learning, we find the number of rounds of random selection to achieve the same classification accuracy. In other words, fixing the classification accuracy achieved, we measure the difference in the training

set size of both methods and report the corresponding training rounds in Table 2. The table shows that active learning achieves a reduction of about 50% in the number of training examples required, i.e., it can reach near optimal performance with 50% fewer training examples. Table 2 reports results for the USPS dataset, however, similar results were obtained for the Pendigits dataset and the Letter dataset.

An important point to note from Table 2 is that active learning does not provide a large benefit in the initial rounds. One reason for this is that all methods start with the same seed set initially. In the first few rounds, the number of examples actively selected are far fewer compared to the seed set size (100 examples). Actively selected examples thus form a small fraction of the total training examples, explaining the small difference in classification accuracy of both methods in the initial rounds. As the number of rounds increase, the importance of active selection becomes clear, explained by the reduction in the amount of training required to reach near-optimal performance.

4.9.3 Time dependence on pool size

From another perspective, the necessity of large active pool sizes points to the importance of computational efficiency in real-world learning scenarios. In order for the methods to be practical, the learning algorithm must be able to select useful images from a huge pool in reasonable time. Empirical data reported in Figure 6 suggests that our method requires time varying linearly with active pool size. While this is much better than traditional methods requiring quadratic or cubic scaling, we improve on this in later parts of the thesis to get sublinear time scaling with respect to active pool sizes.

4.10 Exploring the space

In many applications, the number of categories to be classified is extremely large, and we start with only a few labeled images. In such scenarios, active learning has to balance two often conflicting objectives – exploration and exploitation. Exploration in this context means the the ability to obtain labeled images from classes not seen before. Exploitation refers to classification accuracy on the classes seen so far. Ex-

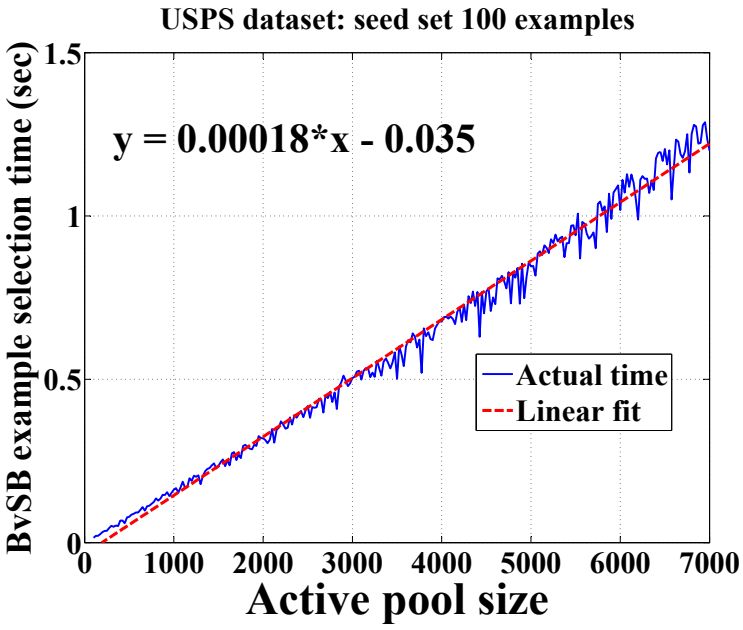


Figure 6: Example selection time as a function of active pool size. The relationship is linear over a large range with the equation shown in the figure. This demonstrates that the method is scalable compared to previous methods requiring quadratic or cubic time. Later in the thesis, we propose approximations that allow sublinear time active selection.

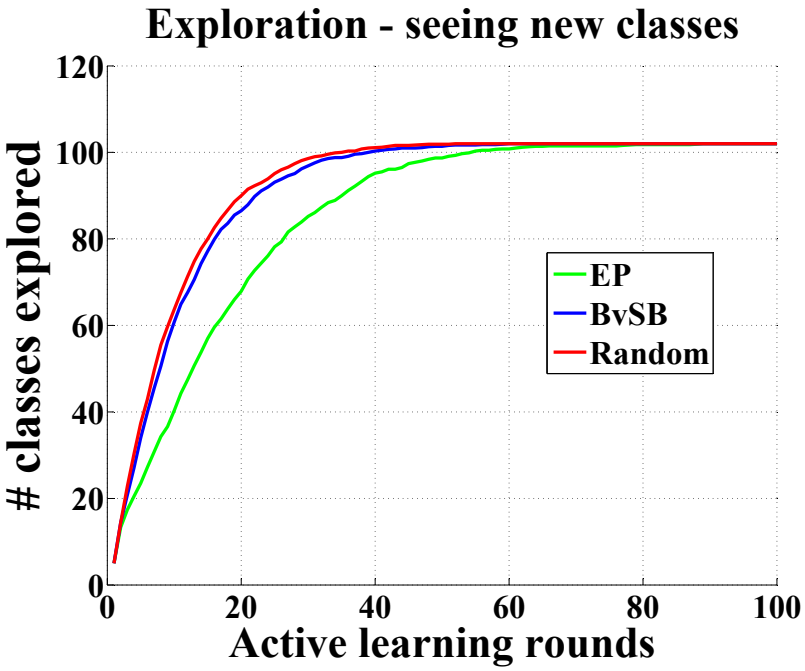


Figure 7: Space exploration of active selection – BvSB-based selection is almost as good as random exploration, while the former achieves much higher classification accuracy than random.

ploitation can conflict with exploration, since in order to achieve high classification accuracy on the seen classes, more training images from those classes might be required, while sacrificing labeled images from new classes. In the results so far, we show classification accuracy on the entire test data consisting of all classes – thus good performance requires a good balance between exploration and exploitation. Here we explicitly demonstrate how the different example selection mechanisms explore the space for the Caltech-101 dataset that has 102 categories. Figure 7 shows that the BvSB measure finds newer classes almost as fast as random selection, while achieving significantly higher classification accuracy than random selection. Fast exploration of BvSB implies that learning can be started with labeled images from very few classes and the selection mechanism will soon obtain images from the unseen classes. Interestingly, EP-based selection explores the space poorly.

4.10.1 Scene recognition

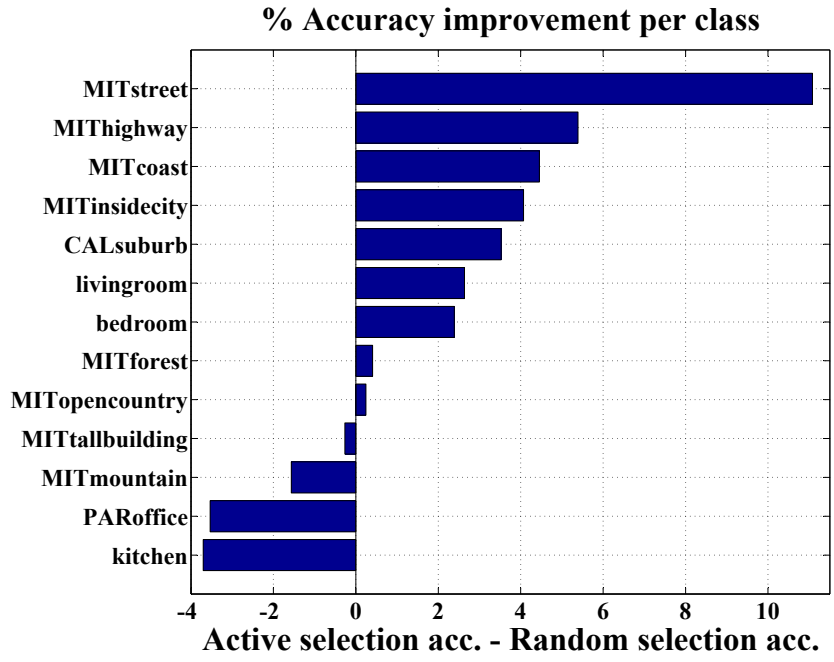


Figure 8: Active learning on the 13 natural scene categories dataset.

Further, we performed experiments for the application of classifying natural scene categories on the 13 scene categories dataset [Fei-Fei and Perona, 2005]. GIST image features of Oliva and Torralba [2001], which provide a global representation were used. Results are shown in Figure 8. The figure shows accuracy improvement (active selection accuracy - random selection accuracy) per class after 30 BvSB-based active learning rounds. Note that although we do not explicitly minimize redundancy amongst images, active selection leads to significant improvements even when *as many as 20 images are selected at each active learning round.*

4.10.2 Which examples are selected?

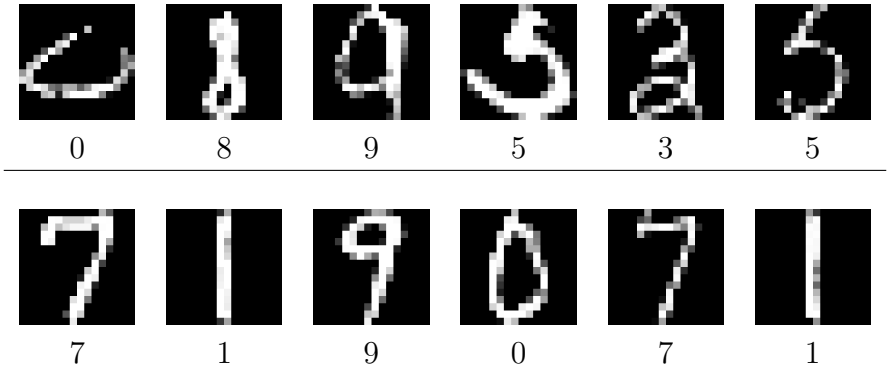


Figure 9: Top row shows images on which the classifier is uncertain using the BvSB score. Bottom row shows images on which the classifier is confident. True labels are noted below the corresponding images. We can see that the top row has more confusing images, indicating that the active learning method chooses harder examples.

In Figure 9, we show example images from the USPS dataset and their true labels. The top row images were confusing for the classifier (indicated by their BvSB score) and were therefore selected for active learning at a certain iteration. The bottom row shows images on which the classifier was most confident. The top row has more confusing images even for the human eye, and ones that do not represent their true label well. We noticed that the most confident images (bottom row) consisted mainly of the digits '1' and '7', which were clearly drawn. The results indicate that the active learning method selects hard examples for query.

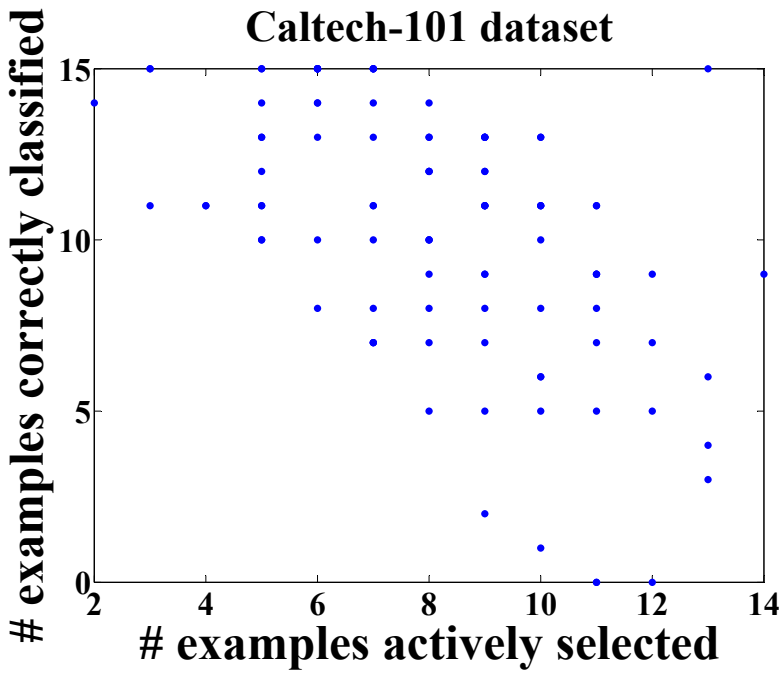


Figure 10: Y-axis: # examples correctly classified by random example selection for a given class. X-axis: # examples of the corresponding class chosen by active selection. The negative correlation shows that active learning chooses more examples from harder classes.

One of the reasons active learning algorithms perform well is the imbalanced selection of examples across classes. In our case, the method chooses more examples for the classes which are hard to classify (based on how the random example selection algorithm performs on them). Figure 10 demonstrates the imbalanced example selection across different classes on the Caltech-101 dataset. On the y-axis, we plot the number of examples correctly classified by the random example selection algorithm for each class, as an indicator of hardness of the class. Note that the test set used in this case is balanced with 15 images per class. On the x-axis, we plot the number of examples selected by the active selection algorithm for the corresponding class from the active pool. The data shows a distinct negative correlation, indicating that more examples are selected from the harder classes, confirming our intuition. Notice the empty region on the bottom left of the figure, showing that active learning selected more images from *all* classes that were hard to classify.

To summarize, the method proposed in this Chapter provides a fast way of uncertainty sampling in multi-class problems. The measure is shown to provided large reductions in required training set sizes even with up to a hundred different classes. In the next section, we utilize this measure for approximating more complex computations of decision theoretic measures for active selection.

5 Multi-Class Learning with Binary Feedback

Even though multi-class active learning methods such as the one proposed in the previous chapter successfully reduce the amount of training data required, they can be labor intensive from a user interaction standpoint for the following reasons: (i) for each unlabeled image queried for annotation, the user has to sift through many categories to input the precise one. Especially for images, providing input in this form can be difficult, and sometimes impossible when a huge (or unknown) number of categories are present; (ii) the time and effort required increase with an increase in the number of categories; (iii) the interaction is prone to mistakes in annotation, and (iv) it is not easily amenable to distributed annotation as all users need to be consistent in labeling.

Image datasets are ever increasing in their size and the image variety - it is not uncommon to have thousands of image classes [Deng et al., 2009; Torralba et al., 2008]. In order to design systems that are practical at larger scales, it is essential to allow easier modes of annotation and interaction for the user. Towards this objective, we propose here a general framework for multi-class active learning that requires only yes/no feedback from the user. A simple illustration of the different interaction models is depicted in Figure 11. During each instance of interaction, the user is presented with two images and has to say whether those images belong to the same category or not. Giving such input is extremely easy, and since only two images need to be compared every time, it is also less prone to human mistakes. It easily allows distributed annotation as well.

5.1 Ease of interaction

In order to quantitatively compare the two interaction modalities, we conducted experiments on 20 users with 50-class and 100-class data, obtained from the Caltech-101 object categories dataset [Fei-Fei et al., 2006]. Each user was asked to interact with

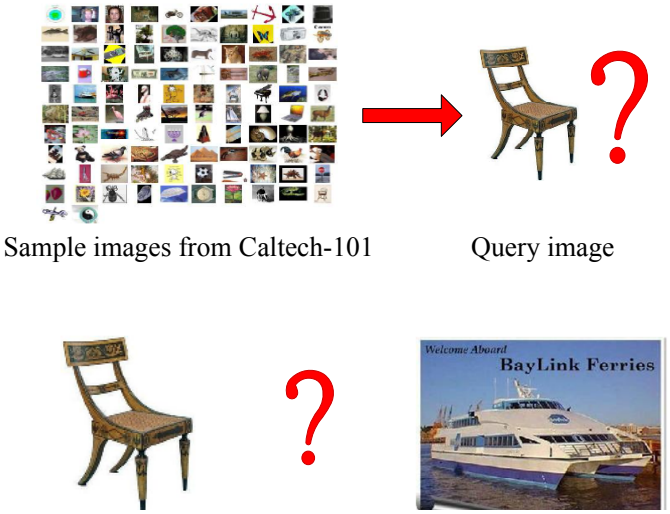


Figure 11: Top row: sample interaction in traditional multi-class active learning approaches. The user needs to input a category name/number for the query image from a large dataset possibly consisting of hundreds of categories. Bottom row: the binary interaction model we propose: the user only needs to say whether or not the query image and the sample image belong to the same category.

two modalities: i) giving category labels (out of a given set of labels) to randomly queried images, as is typically used for training, and ii) giving yes/no responses to two images based on whether they came from the same class. We measured interaction time and the number of errors made in both modalities by each user, along with an overall satisfaction score from 1 through 5, indicating the ease of interaction experienced (1 being the easiest). Table 3 summarizes the results.

Modality	Response time (s)	% errors	Satisfaction
BF – 50 classes	1.6 (±0.2)	0.80	1.2
MCF – 50 classes	11.7 (±3.1)	12.7	4.1
BF – 100 classes	1.7 (±0.2)	0.82	1.1
MCF – 100 classes	28.8 (±5.3)	14.3	4.9

Table 3: Comparing the two interaction modalities.

First, it can be seen that binary feedback (BF) requires far lesser user time than giving multi-class feedback (MCF). Although BF in principle also provides lesser information than MCF, we demonstrate in our experiments that the BF interaction model still achieves superior classification accuracy than MCF with the same expenditure of user time. Second, as seen in the table, MCF has much more noise associated – users make many more errors when sifting through potential categories and finding the correct one. In contrast, BF is much cleaner since it is much easier to simply look at two images and determine whether they belong to the same class or not. Third, the interaction time and annotation errors in MCF increase with the number of categories. This is expected as annotation requires browsing over all possible classes. In contrast, in the BF model, there is no observed increase in user time with increasing number of categories. This aspect is particularly appealing, as the main objective is to scale well to larger problems with potentially thousands of classes. Four, as seen from the satisfaction scores, users are much more satisfied with the overall interaction in BF, since it does not need browsing through many images, and can be done quickly. Apart from the above advantages, distributed annotation across many trainers is easily possible in the BF model. Also, it is straightforward to allow exploration of the data when new categories continuously appear (as opposed to a setting often used previously, wherein the initial training set is created by including examples from all classes [Guo and Greiner, 2007]), or when notions of categories change with time.

In summary, binary feedback provides an extremely appealing interaction model for large problems with many classes. Later sections compare the explicit classification accuracy achieved versus the training time spent in the proposed modality.

5.2 Learning setup

Figure 12 shows a block schematic of the proposed active learning setup. The active pool consists of a large number of unlabeled images from which the active learning algorithm can select images to query the user. The training set consists of images for which category labels are known and can be used for training the classifier.

In the traditional multi-class active learning setting, an unlabeled image (query image) needs to be selected for user annotation. In our case, however, since user

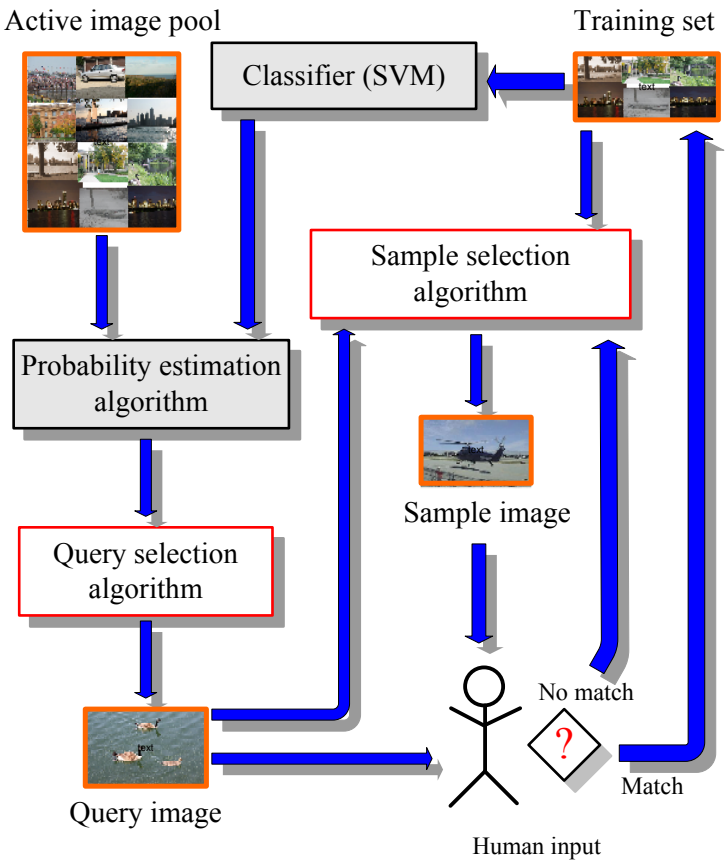


Figure 12: Block schematic of the active learning setting. Our focus in this work is on the query and sample selection algorithms – depicted in white boxes with red borders (see text for details).

input is only binary, we also require an image from a known category to show the user for comparison. Selecting this image from the training set is a new aspect of active selection that our framework requires. We refer to this comparison image from a known category as the "sample image." We focus on query and sample selection algorithms in this chapter – denoted by white boxes with red borders in Figure 12.

Our approach for query as well as sample selection is probabilistic, i.e., based on the current training set, class membership probability estimates are obtained for the images in the active pool. See Chapter 4 for details on multi-class probability estimation based on SVM margins.

In Figure 12, the query selection algorithm selects a query image from the active pool using the estimated class membership probabilities. Based on the estimated membership probabilities for the query image, the sample selection algorithm selects a sample image from the current training set. The query-sample pair is shown to the user for feedback. If a "match" response is obtained, indicating that the query and sample images belong to the same category, the query image is added to the current training set along with its category label. If a "no-match" response is obtained, the sample selection algorithm is again invoked to ask for a different sample image.

This process goes on until either the label for the query image is obtained (with a "match" response), or until the query image does not match any of the categories in the training set. In the latter case, a new category label is initiated and assigned to the query image[5]. Through such a mechanism, the learning process can be started with very few training images initially chosen at random (seed set). As the process continues, the active selection algorithm requires far fewer queries than random selection to achieve similar classification rate on a separate test set. Note that the system is also able to exploit feedback in terms of precise category annotation (as in the typical setting), if available. Binary feedback however generalizes the applicability and allows learning in new unknown environments for exploration.

Binary input has been employed previously in the context of clustering data, by asking the user for pairwise must-link and cannot-link constraints [Basu et al., 2002].

[5]Initiating a new category can require many user responses when many classes are present – we later discuss how to overcome this through a fast new class initialization step along with cluster merging.

This approach can be adapted to the active learning framework by choosing even the sample images from unlabeled data and performing a (unsupervised) clustering step before user annotation. However, in our observation, such an approach was prone to noise due to unsupervised clustering, which can lead to an entire cluster of incorrectly labeled training data. Noise reduction in the preclustering approach is an interesting future work direction. On the other hand, we demonstrate empirically that the setup we employ is robust to labeling noise.

5.3 The active learning method

There are two parts to binary feedback active learning: (i) to select a query image from the active pool, and (ii) to select a sample image from a known category to be shown to the user along with the query image.

5.4 Query selection

The goal here is to query informative images, i.e., images that are likely lead to an improvement in future classification accuracy. We use the Value of Information framework [Krause and Guestrin, 2005; Kapoor et al., 2007b; Vijayanarasimhan and Grauman, 2009] employed in decision theory for query selection here. The broad idea is to select examples based on an objective function that combines the misclassification risk and the cost of user annotation. Consider a risk matrix $M \in \mathbb{R}^{k \times k}$ for a k-class problem. The entry M_{ij} in the matrix indicates the risk associated with misclassifying an image having true label i as belonging to class j. Correct classification incurs no risk and hence the diagonal of M is zero, $M_{ii} = 0, \forall i$.

Denote the estimated class membership distribution for an unlabeled image x as $\mathbf{p}_x = \{p_x^1, \ldots, p_x^k\}$. Note that since the true class membership distribution for x is unknown, the actual misclassification risk cannot be computed – we instead find the *expected* misclassification risk for x as

$$\mathcal{R}_{\mathcal{L}}^{\{x\}} = \sum_{i=1}^{k} \sum_{j=1}^{k} M_{ij} \cdot (p_x^i | \mathcal{L}) \cdot (p_x^j | \mathcal{L}), \tag{8}$$

where $\mathcal{L}$ is the set of labeled examples based on which the probabilities are estimated. Consider that the test set $\mathcal{T}$ consists of N images $x_1, \ldots, x_N$. The total expected risk over the test set (normalized by size) is

$$\mathcal{R}_{\mathcal{L}} = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \sum_{i=1}^{k} \sum_{j=1}^{k} M_{ij} \cdot (p_x^i | \mathcal{L}) \cdot (p_x^j | \mathcal{L}). \tag{9}$$

Note that the above expression requires that the test set be available while computing the total risk. Typically, the test set is not available beforehand, and we can use the images in the active pool $\mathcal{A}$ for computing the expected risk. Indeed, most work on classification uses surrogates to estimate the misclassification risk in the absence of the test set. In many scenarios, the entire available set of unlabeled images is used as the active pool and is typically very large, thus an estimate of risk on the active pool is fairly reliable.

Now, if $y \in \mathcal{A}$ is added to the labeled training set by acquiring its label from the user, the expected reduction in risk on the active pool can be computed as

$$\begin{aligned} \mathcal{R}_{\mathcal{L}} - \mathcal{R}_{\mathcal{L}'} &= \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} \sum_{i=1}^{k} \sum_{j=1}^{k} M_{ij} \cdot (p_x^i | \mathcal{L}) \cdot (p_x^j | \mathcal{L}) \\ &- \frac{1}{|\mathcal{A}'|} \sum_{x \in \mathcal{A}'} \sum_{i=1}^{k} \sum_{j=1}^{k} M_{ij} \cdot (p_x^i | \mathcal{L}') \cdot (p_x^j | \mathcal{L}'), \end{aligned} \tag{10}$$

where $\mathcal{L}' = \mathcal{L} \cup \{y\}$, and $\mathcal{A}' = \mathcal{A} \setminus \{y\}$. The above expression captures the *value* of querying y and adding it to the labeled set. However, we also need consider the *cost* associated with obtaining feedback from the user for y. Assume that the cost of obtaining user annotation on y is given by $\mathcal{C}(y)$. In our framework, we wish to actively choose the image that reduces the cost incurred while maximizing the reduction in misclassification risk. Assuming risk reduction and annotation cost are measured in the same units, the joint objective that represents the value of information (VOI) for a query y is

$$V(y) = \mathcal{R}_{\mathcal{L}} - \mathcal{R}_{\mathcal{L}'} - \mathcal{C}(y). \tag{11}$$

The term $\mathcal{R}_{\mathcal{L}}$ in the above equation is independent of y, the example to be selected for query. Therefore, active selection for maximizing VOI can be expressed as a minimization

$$y^* = \underset{y \in \mathcal{A}}{\operatorname{argmin}} \quad \mathcal{R}_{\mathcal{L}'} + \mathcal{C}(y). \tag{12}$$

Note that the above framework can utilize any notions of risk and annotation cost that are specific to the domain. For instance, we can capture the fact that misclassifying examples belonging to certain classes can be more expensive than others. Such a notion could be extremely useful for classifying medical images so as to determine whether they contain a potentially dangerous tumor. Misclassifying a 'clean' image as having a tumor only incurs the cost of the doctor verifying the classification. However, misclassifying a 'tumor image' as clean could be potentially fatal in a large dataset wherein the doctor cannot manually look at all the data. In such scenarios, the different misclassification risks could be suitably encoded in the matrix M.

As in most work on active learning, our evaluation is based on classification accuracy. As such we employ equal misclassification cost, so that $M_{ij} = 1,$ for $i \neq j$.

5.5 Sample selection

Given a query image, the sample selection algorithm should select sample images so as to minimize the number of responses the user has to provide. In our framework, the sample images belong to a known category; the problem of selecting a sample image then reduces to the problem of *finding a likely category for the query image* from which a representative image can be chosen as the sample image. When presented with a query image and a sample image, note that a "match" response from the user actually gives us the category label of the query image itself! A "no match" response does not provide much information. Suppose that the dataset consists of 100 categories. A "no match" response from the user to a certain query-sample image pair still leaves 99 potential categories to which the query image can belong. Based on this understanding, the goal of selecting a sample image is to maximize the likelihood of a "match" response from the user.

Selecting a sample image (category) can be accomplished by again using the estimated class membership probabilities for the selected query image. For notational simplicity, assume that the query image distribution $\{p_1, \ldots, p_k\}$ is in sorted order such that $p_1 \geq p_2 \geq \ldots \geq p_k$. The algorithm proceeds as follows. Select a representative sample image from class 1 and obtain user response. As long as a "no match" response is obtained for class $i-1$, select a sample image from class i to present the user. This is continued until a "match" response is obtained. Through such a scheme, sample images from the more likely categories are selected earlier in the process, in an attempt to minimize the number of user responses required.

5.5.1 Annotation cost

In the binary feedback setting, our experiments indicated that it is reasonable to assume that each binary comparison requires a constant cost (time) for annotation. Thus, for each query image, the cost incurred to obtain the class label is equal to the number of binary comparisons required. Since this number is unknown, we compute its expectation based on the estimated class membership distribution instead. If the distribution is assumed to be in sorted order as above, the expected number of user responses to get a "match" response is

$$\mathcal{C}(x) = p_1^x + \sum_{j=2}^{k} (1 - p_1^x) \ldots (1 - p_{j-1}^x) \cdot p_j^x \cdot j, \tag{13}$$

which is also the user annotation cost. We can scale the misclassification risk (by scaling M) with the real-world cost incurred to find the true risk, which is in the same units as annotation cost. Here we choose the true risk as the *expected number of misclassifications* in the active pool, and compute it by scaling M with the active pool size. Along with our choice of $\mathcal{C}(x)$, this amounts to equating the cost of each binary input from the user to every misclassification, i.e., we can trade one binary input from the user for correctly classifying one unlabeled image.

5.6 Stopping criterion

The above VOI-based objective function leads to an appealing stopping criterion – we can stop whenever the maximum expected VOI for any unlabeled image is **negative**, i.e., $\text{argmax}_{x \in \mathcal{A}} V(x) < 0$. With our defined notions of risk and cost, negative values of VOI indicate that a single binary input from the user is not expected to reduce the number of misclassifications by even one, hence querying is not worth the information obtained. It should be noted that different notions of real-world risk and annotation cost could be employed instead if specific domain knowledge is available. The selection and stopping criteria directly capture the particular quantities used.

5.7 Initiating new classes

Many active learning methods make the restrictive assumption that the initial training set contains examples from all categories [Guo and Greiner, 2007]. This assumption is unrealistic for most real problems, since the user has to explicitly construct a training set with all classes, defeating our goal of reducing supervision. Also, if a system is expected to operate over long periods of time, handling new classes is essential. Thus, we start with small seed sets, and allow dynamic addition of new classes. In the sample selection method described above, the user is queried by showing sample images until a "match" response is obtained. However, if the query image belongs to a category that is not present in the current training set, many queries will be needed to initiate a new class.

Instead, we initiate a new class when a fixed small number (say 5) of "no-match" responses are obtained. With good category models, the expected distributions correctly capture the categories of unlabeled images – hence, "no-match" responses to the few most likely classes often indicates the presence of a previously unseen category. However, it may happen that the unlabeled image belongs to a category present in the training data. In such cases, creating a new class and assigning it to the unlabeled image results in overclustering. This is dealt with by agglomerative clustering (cluster merging), following the min-max cut algorithm [Ding et al., 2001], along with user input.

The basic idea in agglomerative clustering is to iteratively merge two clusters

Input: Labeled set $\mathcal{L}$, active pool $\mathcal{A}$, cost matrix M

1. $\mathcal{L}^0 := \mathcal{L}$; $\mathcal{A}^0 := \mathcal{A}$
2. **for** round $r = 0$ **to** $n - 1$ **do**
3. **foreach** image $x_i \in \mathcal{A}^{(r)}$ **do**
4. **for** class $y_i = 1$ **to** k **do**
5. Train multi-class classifier with $\mathcal{L}^{(r)} \cup \{x_i, y_i\}$
6. Estimate class membership probabilities for images in the active pool $\mathcal{A}^{(r)}$
7. Compute risk on the active pool $R^{(x_i,y_i)}$
8. **end**
9. Compute expected risk ($\mathcal{L}'^r = \mathcal{L}^r \cup \{x_i\}$) $\mathcal{R}_{\mathcal{L}'^r} = \sum_l P(y_i = l) \cdot R^{(x_i,l)}$
10. Compute expected annotation cost $\mathcal{C}(x_i)$
11. **end**
12. Find image $x^* = \text{argmin}_{x_i \in \mathcal{A}^{(r)}} \mathcal{R}_{\mathcal{L}'^r} + \mathcal{C}(x_i)$
13. Find $V(x^*)$ using Eqn. (11)
14. **if** $V(x^*) > 0$ **then**
15. Query user with query image x^* and likely sample images until true label k^* is obtained
16. Set $\mathcal{L}^{(r+1)} := \mathcal{L}^{(r)} \cup \{x^*, k^*\}$; and $\mathcal{A}^{(r+1)} := \mathcal{A}^{(r)} \setminus \{x^*\}$
17. **else** return $\mathcal{L}^{(n)} = \mathcal{L}^{(r)}$
18. **end**

Output: The new labeled set $\mathcal{L}^{(n)}$

Figure 13: Multi-class active learning with binary feedback.

that have the highest similarity (linkage value) $l(C_i, C_j)$. For min-max clustering the linkage function is given by

$$l(C_i, C_j) = s(C_i, C_j)/(s(C_i, C_i)s(C_j, C_j)),$$

where s indicates a cluster similarity score

$$s(C_i, C_j) = \sum_{x \in C_i} \sum_{y \in C_j} K(x, y).$$

Here K is the kernel function that captures similarity between two objects x and y (the same kernel function is also used for classification with SVM).

In our algorithm, we evaluate cluster linkage values after each iteration of user feedback. If the maximum linkage value (indicating cluster overlap) is for clusters C_i and C_j, and is above a threshold of 0.5, we query the user by showing two images from C_i and C_j. A "match" response results in merging of the two clusters. Note that our setting is much simpler than the unsupervised clustering setting since we **have user feedback available**. As such, the method is relatively insensitive to the particular threshold used, and lesser noise is encountered. Also, note that we do not need to compute the linkage values from scratch at each iteration – only a simple incremental computation is required. In summary, new classes are initiated quickly, and erroneous ones are corrected by cluster merging with little user feedback.

5.8 Computational considerations

The computational complexity of each query iteration in our algorithm (Figure 13) is $\mathcal{O}(n^2k^3)$, with an active pool of size n and k classes. Although it works well for small problems, the cost can be impractical at larger scales. Instead of performing the exact computation however, we compute a multi-class margin style approximation as defined in the previous chapter.

5.8.1 Approximations to VOI

In the previous chapter, we showed how the proposed BvSB uncertainty sampling measure can efficiently select informative examples for active learning. Here we discuss some approximations that substantially improve the running time of the proposed VOI algorithm using the BvSB measure. The VOI algorithm described previously in Figure 13 is the original algorithm on which the following approximations are performed (line numbers refer to the algorithm).

5.8.2 Seed sampling

Since VOI computation is relatively expensive, finding the scores for all examples in the active pool is costly (line 3). Instead, we use the BvSB measure to sample uncertain examples from the active pool on which VOI computation is performed. Typically, a sample of 50 examples is obtained from active pools of thousands of examples. We observed that even though BvSB and VOI do not correlate perfectly, the the top 50 examples chosen by BvSB almost always contain the examples that would have been the highest ranked using VOI alone. Quantitatively, the results differ only 2% of the time, and the difference in classification accuracy is negligible. On the other hand, the computational speedups achieved are substantial.

5.8.3 Expected value computation

In the VOI algorithm, estimating expected risk is expensive. For each unlabeled image, we need to train classifiers assuming that the image can belong to any of the possible categories (line 4). This can be slow when many classes are present. To overcome this, we make the following observation: given the estimated probability distribution of an unlabeled image, it is unlikely to belong to the classes that are assigned low probability values, i.e., the image most likely belongs to the classes that have the highest estimated probabilities. As such, instead of looping over all possible classes, we can only loop over the most likely ones. In particular, we loop over only the top 2 most likely classes, as they contain most of the discriminative information, as utilized in the BvSB measure. Such an approximation relies to some extent on the correctness of the estimated model, which implies an *optimistic* assumption often

made for computational tractability [Guo and Greiner, 2007]. Further, we can use the same "top-2" approximation, for computing the expected risk (line 9) on unlabeled images, as an approximation to Eqn. (8).

5.8.4 Clustering for estimating risk

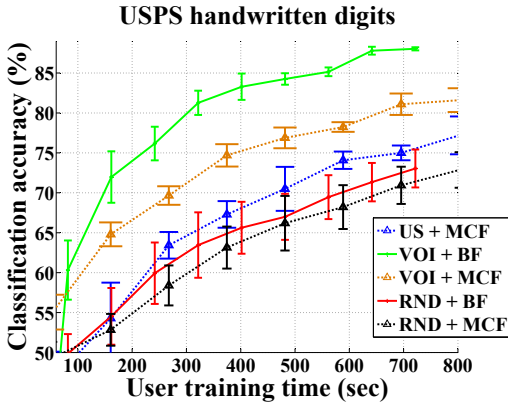
In the above algorithm, the risk needs to be estimated on the entire active pool. Instead, we first cluster the unlabeled images in the active pool using the kernel k-means algorithm [Shawe-Taylor and Cristianini, 2004]. Then we form a new unlabeled image set by choosing one representative (closest to the centroid) image from each cluster, and estimate risk on this reduced set. The clustering needs to be performed only once initially, and not in every query iteration. In our implementation, we fix the number of clusters as 1/100 fraction of the active pool size. Experiments showed that this approximation rarely (less than 5% of the time) changes the images selected actively, and makes a negligible difference in the estimated risk value, and the future classification accuracy.

With the above approximations, the complexity of each query iteration is $\mathcal{O}(nk^2)$, a large improvement over the original version. Later chapters further improve the running time to sublinear in the pool size with the help of approximate near neighbor search algorithms.

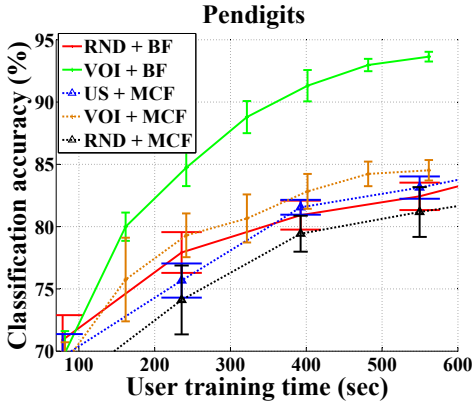
5.9 Experimental results

In this section, we evaluate the proposed VOI algorithm on various datasets described in Table 1. Scene-13 is a dataset of 13 natural scene categories [Fei-Fei and Perona, 2005], for which we employ GIST features [Oliva and Torralba, 2001]. Precomputed pyramid match kernel matrices [Grauman and Darrell, 2005] were used as features for the Caltech-101 dataset.

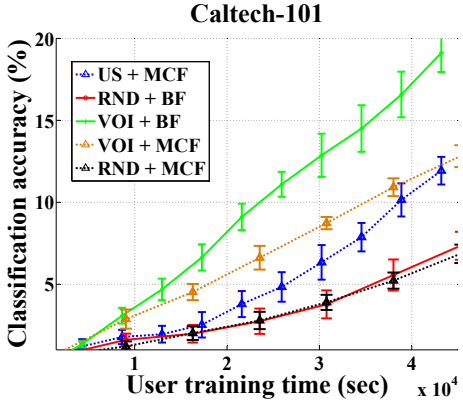
For implementation we used Matlab along with the LIBSVM toolbox [Chang and Lin, 2001] (written in C, interfaced with Matlab for SVM and probability estimation). With an active pool size of 5000 images for a 10-class problem (USPS) each query iteration on average takes about 0.9 seconds on a 2.67 Ghz Xeon machine. For the Caltech dataset with an active pool of size 1515 images with 101 classes, a query



(a)



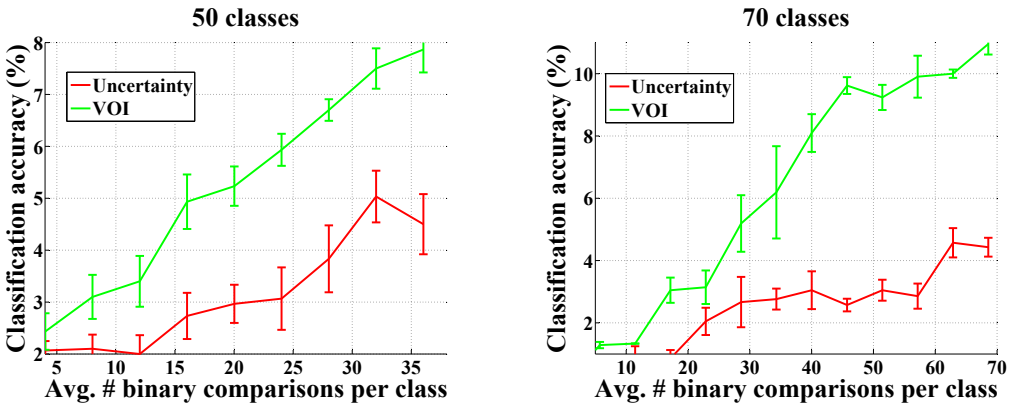
(b)



(c)

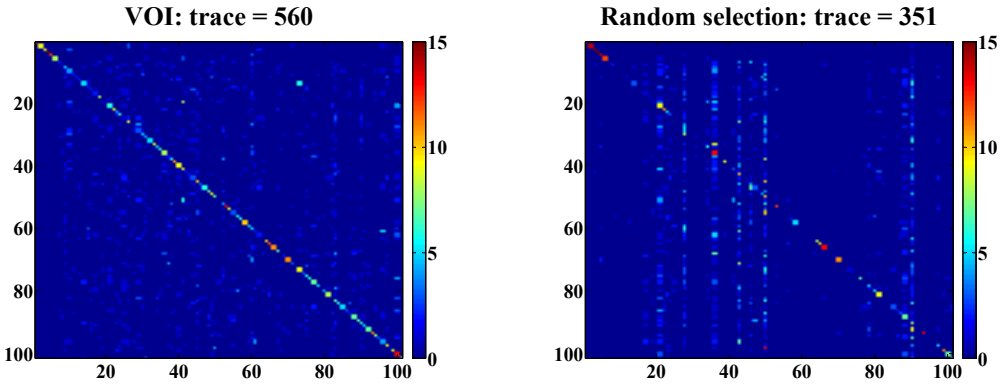
Figure 14: Active learning in the BF model requires far lesser user training time compared to active selection in the MCF model. US: uncertainty sampling, RND: random. (a) USPS, (b) Pendigits, (c) Caltech-101 datasets.

iteration takes about 1.3 seconds.



(a) (b)

Figure 15: VOI-based active selection and uncertainty sampling (both with BF) during the initial phases of active learning.



(a) (b)

Figure 16: Confusion matrices with (a) active (VOI), and (b) random selection (max. trace = 1515). VOI leads to much lower confusion.

5.10 User interaction time

We have previously demonstrated the benefits of the BF model as compared to MCF from the ease of interaction standpoint. Here we compare the total user annota-

tion time required with various methods to achieve similar classification rates. The comparison shows the following methods: our proposed VOI method with binary feedback (VOI+BF), VOI with Multi-class feedback (MCF), active learning using only the BvSB measure (US+MCF), where US stands for uncertainty sampling, and random selection with both BF and MCF. Figure 14 shows the substantial reduction in user training time with the proposed method. For all the datasets, the proposed VOI-based algorithm beats all others (including active selection with MCF), indicating that the advantages come from both **our active selection algorithm, as well as the binary feedback model**. Further, note that the relative improvement is larger for the Caltech dataset, as it has a larger number of categories. As such, we can train classifiers in a fraction of the time typically required, demonstrating the strength of our approach for multi-class problems.

5.11 Importance of considering annotation cost

As mentioned before, we use uncertainty sampling(US)-based active selection to form a smaller set from which the most informative images are selected using VOI computation. Here we demonstrate that the good results are not due to uncertainty sampling alone. Figure 15 compares the *number of binary comparisons the user has to provide* in our algorithm along with the BvSB uncertainty sampling method (also in the BF model) in the initial stages of active learning. The figure shows two plots with 50 and 70 class problems, obtained from the Caltech-101 dataset. Our method significantly outperforms US in both cases, and the relative improvement increases with problem size. As the number of classes increases, considering user annotation cost for each query image becomes increasingly important. The VOI framework captures annotation cost unlike US, explaining the better performance for the 70 class problem.

5.12 Active selection (VOI) vs. random selection

Figure 16 shows the confusion matrices for active selection with VOI as well as random selection on the Caltech 101 class problem. Active selection results in much lesser confusion, also indicated by the trace of the two matrices. This demonstrates that the

algorithm offers large advantages for many category problems. Figure 18 shows per-class classification accuracy of both VOI and random selection methods on the Scene-13 dataset. VOI achieves higher accuracy for 9 of the 13 classes, and comprehensively beats random selection in the overall accuracy.

5.13 Noise sensitivity

In many real-world learning tasks, the labels are noisy, either due to errors in the gathering apparatus, or even because of human annotation mistakes. It is therefore important for the learning algorithm to be robust to a reasonable amount of labeling noise. In this section, we perform experiments to quantify the noise sensitivity of the methods. We artificially impart stochastic labeling noise to the training images. For example, 5% noise implies that training images are randomly given an incorrect label with a probability of 0.05. The algorithms are then run on the noisy as well as clean data – results for the USPS dataset are shown in Figure 17.

The figure shows both active and random selection on clean as well as noisy data (10% and 20% noise). Expectedly, there is a reduction in classification accuracy for both algorithms when noise is introduced. Interestingly, however, even with as much as 10% label noise, the active learning method still outperforms random selection on clean data, whereas with about 20% noise, active learning still matches random selection on clean data. This result shows that active selection can tolerate a significant amount of noise while giving a high classification rate.

One reason why active selection can be robust to noise arises from the fact that the algorithm selects "hard" examples for query. In most cases, these examples lie close to the separating boundaries of the corresponding classifiers. Intuitively, we expect noise in these examples to have a smaller effect, since they change the classification boundary marginally. In contrast, a misclassified example deep inside the region associated with a certain class can be much more harmful. In essence, through its example selection mechanism, active learning encounters noise that has a relatively smaller impact on the classification boundary, and thus the future classification rate.

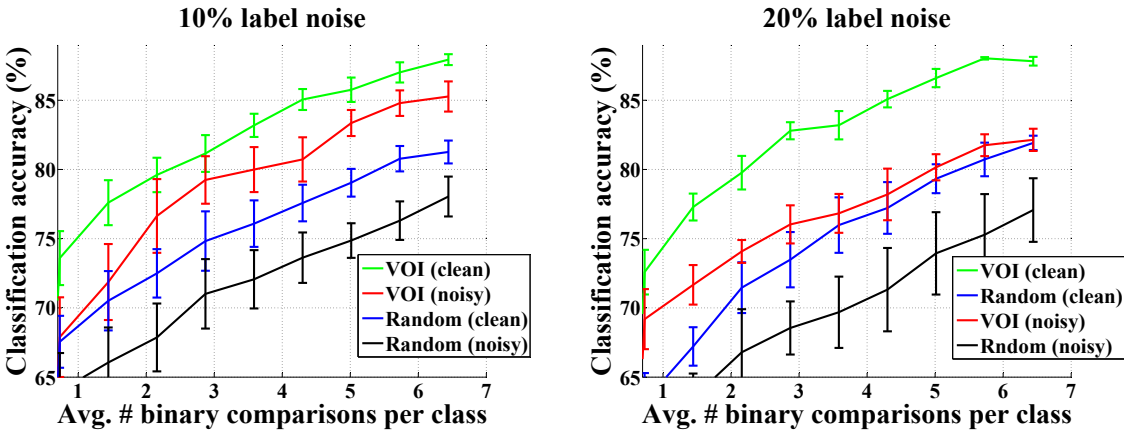


Figure 17: Sensitivity to label noise, (a) 10%, (b) 20%. VOI with noisy data outperforms the random selection with clean data.

5.14 Population imbalance

Real-world data often exhibits class population imbalance, with vastly varying number of examples belonging different classes [Ertekin et al., 2007]. For example, in the Caltech-101 dataset, the category 'airplanes' has over 800 images, while the category 'wrench' has only 39 images.

We demonstrate here that active selection can effectively counter population imbalances in order to generalize better. The experiment is conducted as follows. The active pool (from which unlabeled images are selected for query) consisting of vastly varying number of examples of each class is generated for the Pendigits dataset. However, the test set is kept unmodified. In this scenario, random example selection suffers since it obtains fewer examples from the less populated classes. Active selection, on the other hand, counters the imbalance by selecting a relatively higher number of examples even from the less populated classes. Figure 19 demonstrates the results. The three bars show (normalized) numbers of examples per class in the unlabeled pool, and in the training sets with active and random selection. Random selection does poorly – for instance, it does not obtain even a single training image from class '9' due to its low population in the unlabeled pool. Active selection overcomes population imbalance and selects many images from class '9'. This is further

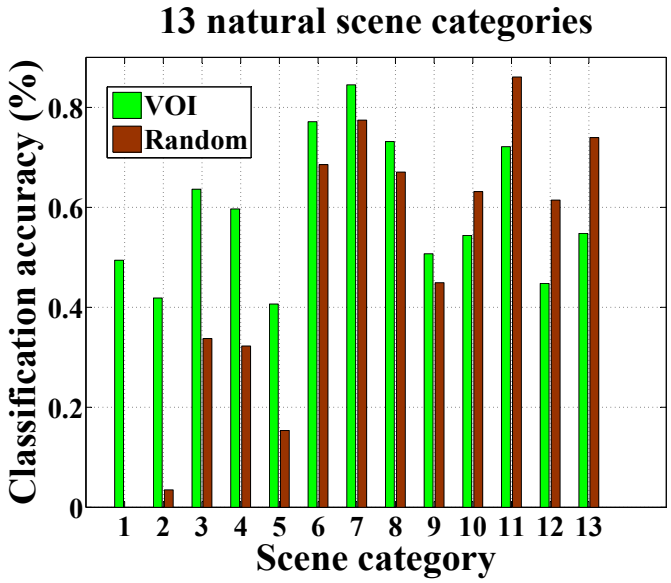


Figure 18: Per-class accuracy of VOI v/s random on the scene-13 dataset.

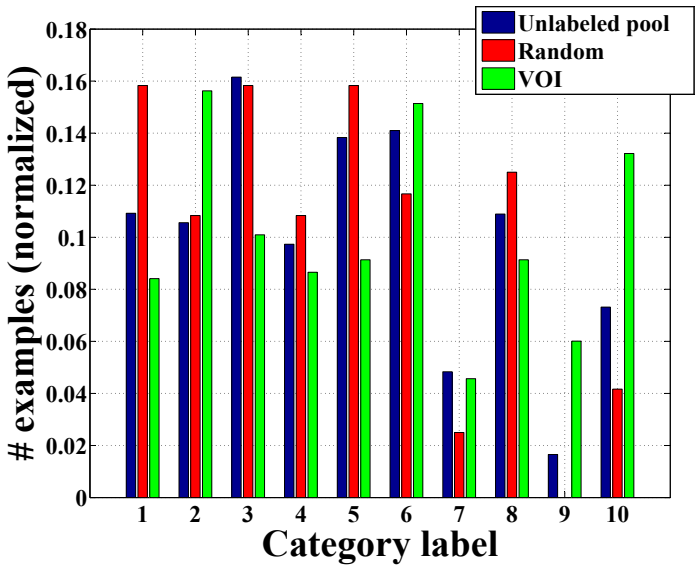


Figure 19: Population imbalance: VOI selects many images even for classes with small populations (see text for details).

reinforced by computing the variance in the normalized population. The standard deviation in the (normalized) number of examples selected per class with active and random selection is 0.036 and 0.058 respectively. The significantly smaller deviation shows that active selection overcomes population imbalance to a large extent.

5.15 Fast initiation of new classes

Dataset	W/ clustering	Naive
Caltech-101	2560 sec	3200 sec

Table 4: User training time required to encounter all 101 classes.

In Section 5.7, we described our method of quickly initiating new classes and then merging the erroneous ones using agglomerative clustering and user feedback. Table 4 summarizes the advantages of the approach (i.e., w/ clustering) compared to simple category initiation when a new image does not match any training image (naive). We start with a small seed set of 20 images, and run the experiment until both methods encounter all the 101 categories in the data. Note the large reduction in user training time with clustering, due to the fewer number of binary comparisons requested. This aspect is increasingly important as the number of classes increases.

In summary, we presented a new multi-class active learning framework that requires only binary feedback from the user. Experiments on large datasets demonstrated the benefits of our approach, in terms of substantially reducing user training time and effort. The proposed method was also shown to be robust to real-world issues such as population imbalance and noise. In the following chapters, we focus on scaling the active learning method to deal with very large data sizes, while still maintaining the classification rates. Even though the binary interaction modality is not explicitly mentioned henceforth, note that it can be used with any formulation that requests annotations from a user.

6 Scaling up Active Learning

There has been some recent work on scaling up active learning to work with large datasets. In [Zhao et al., 2008], a graph-regularization approach is proposed to maximize the expected information gain for scalable active learning. Segal et al. [2006] propose an approximate uncertainty sampling approach in which only a subset of samples are evaluated at each iteration for active learning. Their approach provides speedups for the application of labeling email corpora. A hierarchical sampling approach along with feature space indexing was proposed for scaling active learning to large datasets by Panda et al. [2006].

In this section, we show initial results with a different approach to speeding up active learning via locality sensitive hashing (LSH) [Indyk and Motwani, 1998]. As opposed to previous work, our method does not modify the active selection criteria and can work with any classifiers. Instead of performing an exhaustive search with a linear scan (LS) of the entire unlabeled pool, the main idea is to first find representative samples that are informative (for seeding the search) according to our active selection measure. Using locality sensitive hashing on these samples, informative samples from the unlabeled pool are obtained (in time scaling sublinearly with the pool size). This approach provides *3-4 orders of magnitude speed-up* on the linear scan active learning version, while making little difference in classification accuracy. We can thus scale the algorithm to datasets with hundreds of thousands of samples. With a pool size of 50000 images represented in 384-dimensional space, the LSH-based approximation provides a *91-fold speedup on average* with negligible reduction in classification accuracy. In the following, we provide a brief introduction to LSH using p-stable distributions [Datar et al., 2004] followed by its application in our active learning algorithm.

6.1 LSH with p-Stable Distributions

Consider a d-dimensional space $\mathbb{R}^d$, with the p-norm denoted by $||v||_p^d$ for vector v. Let the metric space be $\mathcal{M} = (X, d)$, in which the ball of radius r centered at q is defined as $B(q, r) = \{v \in X \mid d(v, q) \leq r\}$.

Given a dataset P and a query q, in the (R, c)–near neighbor (NN) problem [Indyk and Motwani, 1998], one has to retrieve points p such that $d(p, q) \leq cR$, if there exists a point in P within distance R from q. In other words, the approximate nearest neighbors retrieved must be bounded close to the true nearest neighbor.

Definition

[Indyk and Motwani, 1998] A LSH family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2)-sensitive for D if for any $u, v \in S$,

- if $u \in B(v, r_1)$, then $Pr_{\mathcal{H}}[h(u) = h(v)] \geq p_1$,
- if $u \notin B(v, r_2)$, then $Pr_{\mathcal{H}}[h(u) = h(v)] \leq p_2$.

If $p_1 > p_2$ and $r_1 < r_2$, the family $\mathcal{H}$ can be used for the (R,c)-NN problem. The basic idea is that the hash functions evaluate to the same values with high probability for points that are close to each other, whereas for distant points the probability of matching (collision) is low. The probability gap can be increased by concatenation of multiple hash functions chosen randomly from the family $\mathcal{H}$.

Using the notation in Datar et al. [2004], define the function family $\mathcal{G} = \{g : S \to U^k\}$, where $g(v) = \{h_1(v), \ldots, h_k(v)\}$, where $h \in \mathcal{H}$. For a given L, choose $g_1, \ldots, g_L$ uniformly at random from $\mathcal{G}$. k and L can be chosen to satisfy the desired collision probability guarantees as described in the next section.

In the pre-precessing step, each data sample from the dataset is stored in buckets $g_i(x), i \in \{1, \ldots, L\}$. For a given query q, points from all the buckets $g_i(q), i \in \{1, \ldots, L\}$ are retrieved. The nearest neighbor from these retrieved points is then returned as the approximate nearest neighbor. It is shown in [Datar et al., 2004] that given a (R, cR, p_1, p_2)–sensitive family $\mathcal{H}$ for the distance measure d, then there exists an algorithm that solves the (R, c)-NN problem in query time $\mathcal{O}(N^\rho)$, where N is the

dataset size and $\rho = \frac{ln1/p_1}{ln1/p_2}$. Thus for $p_1 > p_2$, the above results in sublinear time retrieval.

It is further shown by Datar et al. [2004] that the hash functions $h_{\mathbf{a},b}(\mathbf{x}) = \lfloor \frac{\mathbf{a}\cdot\mathbf{x}+b}{r} \rfloor$, where each element of $\mathbf{a}$ is sampled from $\mathcal{N}(0,1)$, and b chosen uniformly from $[0,r]$ represents a (R, cR, p_1, p_2)-sensitive LSH family for the Euclidean distance measure. The result crucially relies on the fact that the Gaussian distribution from which $\mathbf{a}$ is sampled is a 2-stable distribution. As such, we use this hash family for fast search of informative samples in our active learning algorithm.

6.2 Choice of parameters

In the standard setting, the input to the algorithm is the parameter $c = (1+\epsilon)$, which provides a $\epsilon-$approximate nearest neighbor with high probability. It is shown in [Datar et al., 2004] that for the case of $p = 2$ (Gaussian distribution), $p_2 = 1-2\cdot normcdf(-r/c)-\frac{2}{\sqrt{2\pi}r/c}(1-e^{(-r^2/2c^2)})$ and $p_1 = 1-2\cdot normcdf(-r)-\frac{2}{\sqrt{2\pi}r}(1-e^{(-r^2/2)})$, where $normcdf(\cdot)$ is a cumulative distribution function for a zero-mean, unit-variance Gaussian. Given c, the goal is to find r to achieve the optimal ρ. They further show experiments demonstrating that the value of ρ is not very sensitive to changes in r as long as it is not "too small." Based on their plots and experiments, we fix $r = 4$ for all the results below.

The value of k represents a tradeoff between time to compute the hash and the number of false positives. A lower value of k results in fast hash computation but higher false positive rate. We use $k = 30$, and $L = 5$ in all our experiments since these values provide good results in practice.

6.3 Sublinear time Active Learning

Here we propose a simple way to speed up active learning using LSH. During preprocessing, we first hash all the points in the database to the respective buckets using the chosen hash functions. At each iteration, we pick the samples from our training data that give the highest VOI assuming they are unlabeled. These samples are treated as *informative seed samples* that will be used as queries to retrieve the nearest neighbors from the active pool, in the hope that they will also be informative. Since the training

set is usually orders of magnitude smaller than the unlabeled pool, a linear scan to choose best samples from it does not slow down the algorithm. Also, other seeding strategies that do not require a scan could easily be employed instead.

Assuming that the VOI function is spatially smoothly, the rationale behind choosing the nearest neighbors of the points with high VOI is to find other *unlabeled points with high VOI*. Intuitively, many functions that capture informativeness of samples such as distance to hyperplane etc. can be reasonably assumed to be smooth, so that such a search will lead to useful samples for active learning. Furthermore, note that the proposed strategy does not depend on the choice of the classifier or the active selection measure used. It can be employed for other classifiers as well as other selection measures seamlessly. The hashing method as proposed requires the explicit feature vectors of the data samples, and as such cannot be used directly for kernel matrices. Extending to kernels will be an interesting direction for future work.

The following section shows results using the proposed speedup technique compared to performing a linear scan of the entire unlabeled pool.

6.4 Experiments with hashing

Experiments are performed on two datasets: the USPS dataset used previously, and the Cifar-10 dataset [Krizhevsky, 2009], which is a collection of 50000 training images and 10000 test images obtained from the 80 million tiny images dataset [Torralba et al., 2008]. For Cifar-10, 384-d GIST descriptors [Oliva and Torralba, 2001] are used as per Krizhevsky [2009].

Our algorithm relies on LSH retrieving points that are close to the query points with high probability. Here we first perform an experiment with the Cifar-10 dataset to analyze how efficiently nearest neighbors are retrieved by LSH. The setup is as follows. For each iteration, a random point was selected as the query. The LSH and LS were run to find the near neighbors of the query, while noting the time required for both along with the distance to the nearest neighbor found (LS finds the true nearest neighbor). The distance to the nearest neighbor found by LSH is normalized by the distance to the true neighbor to find the approximation factor $c = 1 + \epsilon$. We ran 1000 such iterations and the resulting speedup values were put into 5 bins.

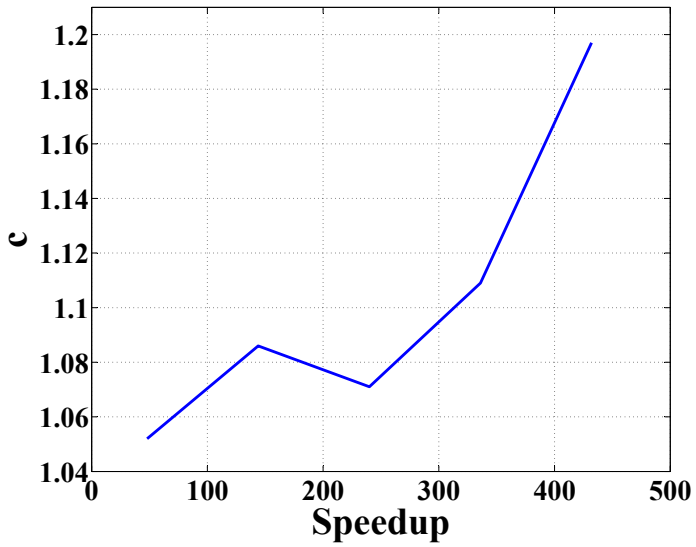


Figure 20: Speedup achieved with LSH over LS for the approximate near neighbor problem on the Cifar-10 dataset. $c = 1 + \epsilon$ denotes the approximation factor.

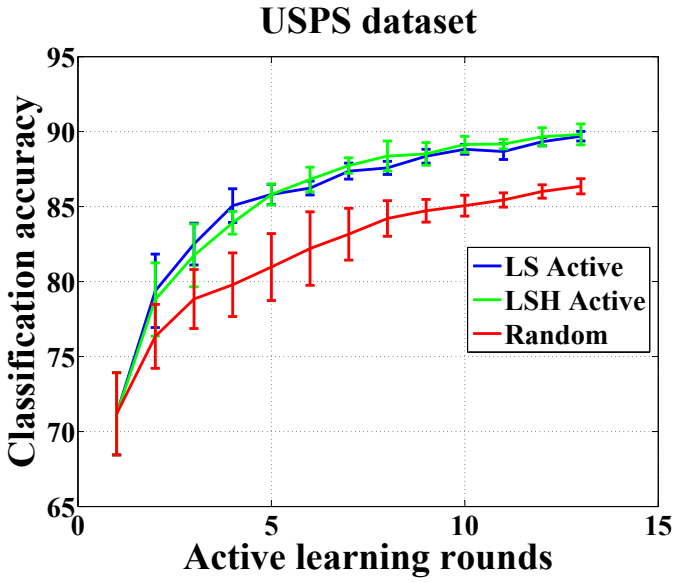


Figure 21: Active learning with the LSH approximation gives little difference in accuracy compared to Linear Scan on the USPS dataset. Speedup achieved over linear scan was 17-fold.

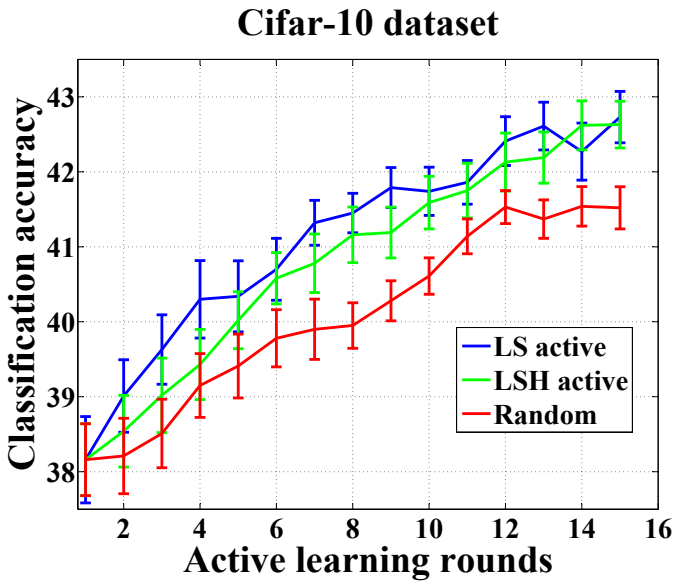


Figure 22: Results on the Cifar-10 dataset. The improvement due to active learning is smaller, as this is a more challenging classification task, however, LSH still provides a close approximation. The speedup achieved over linear scan was 91-fold.

Figure 20 shows a plot of the approximation factor achieved versus speedup. As expected, we see that a higher speedup gives worse approximation. The speedups however are large across the entire spectrum of approximation values, achieving a 400-fold speedup for a 0.2-approximation ($c = 1.2$). Note that the approximation guarantees for LSH are conservative, and we observe significantly better performance in practice. Furthermore, since the LSH algorithm scales sublinearly with data size, we expect the speedups to be even larger for bigger datasets.

It is important to note that even a crude approximation to nearest neighbor does not necessarily hurt active learning. Active selection measures are typically based on computations of potential informativeness of the data sample which are often approximate, and are heavily dependent on the current model. As such, even points that are not the nearest neighbors to informative queries might have very close (and sometimes even better) informativeness scores than the true nearest neighbors. Our experiment below demonstrates that this is indeed the case: an approximate nearest neighbor often makes no difference in the informativeness values of the chosen samples

as well as in the final classification accuracy achieved.

Figures 21 and 22 show classification accuracy comparisons between LS and LSH active learning algorithms. In both plots, the difference in accuracy due to the approximation is very small, whereas the LSH-based active learning algorithms run about 1 and 2 orders of magnitude faster respectively on USPS (∼5000 samples) and Cifar-10 (∼50000 samples). As mentioned before, the speedup is expected to increase with the dataset size, since the linear scan takes $\mathcal{O}(n)$ time (for a pool size of n) whereas LSH-based active learning runs in expected time $\mathcal{O}(n^\rho)$ with $\rho < 1$. This demonstrates the powerful scaling ability of the locality sensitive hashing approach to active learning.

Note that in some cases, it is possible that nearest neighbors to informative samples in the training set might not be informative for active selection. In such cases, one can continue searching for nearest neighbors of the points found in the previous iteration (and the uninformative ones according to the selection measure removed). In such a way, a larger region of the space is explored, progressively moving towards more informative samples at the cost of a higher number of point scanned. However, we observed from our experiments that this was not essential for most of the datasets. In one case, we used 2 iterations instead of 1 for improved results, however the difference was small, and might not be worth the extra computation required.

The described LSH method can be used with vectors in Euclidean space, Hamming space, etc. LSH techniques have also been proposed for cosine similarities in similar document retrieval applications. For images, region covariance descriptors are powerful tools for image matching. We describe in the following how a similar approximate nearest neighbor search can be applied to region covariance descriptors, while respecting the implicit distance measure (i.e., without naively vectorizing the descriptors).

6.5 LSH for Covariance Matrices

Region covariance descriptors [Tuzel et al., 2008] are popular feature descriptor for various image classification applications, e.g., pedestrian detection, face recognition, probabilistic tracking, etc. Suppose that each pixel in the image is represented as

a d-dimensional feature vector x incorporating features such as color, edges, etc., depending on the application requirements. A region of the image R (an image patch) can then be represented by a d x d covariance matrix C of the vectors $\{x_i\}_{i=1}^{|R|}$.

The space of all such region covariance descriptors forms a connected Riemannian manifold, and geodesic distances over the manifold capture similarity between two descriptors. The geodesic distance between two covariance descriptors C_i and C_j is given by the length of the geodesic connecting C_i and C_j and can be computed as [Pennec et al., 2006]:

$$d_{\mathcal{G}}(C_i, C_j) = \| \log(C_i^{-1/2} C_j C_i^{-1/2}) \|_F, \tag{14}$$

where the subscript $\mathcal{G}$ denotes geodesic distance on the manifold, and log is the matrix logarithm.

Region covariance descriptors are often vectorized (typically in a row-major scan of the upper triangular part, since they are symmetric) in order to apply standard vector-space learning algorithms. The primary limitation of this approach is that geodesic structure is lost due to the vectorization. Since vectorization does not respect geodesic distances, notions of similarity of the region covariance descriptors cannot not captured by relying of vector data.

We are also not aware of any locality-sensitive hashing approaches for the Riemannian distance metric. In this section, we propose an locality sensitive hashing scheme that approximately respects geodesic distances on the manifold, while at the same time offers probabilistic guarantees similar to the hashing in Euclidean space. The main idea is to utilize the Log-Euclidean distance that lower bounds the Euclidean distance (and often provides a close approximation). The matrix logarithm is an embedding of the region covariances into Euclidean space; the distance metric induced is given by [Arsigny et al., 2006]:

$$d_{\mathcal{LE}}(C_i, C_j) = \| \log C_i - \log C_j \|_F, \tag{15}$$

where $\mathcal{LE}$ stands for the Log-Euclidean distance. Bhatia [2007] shows that the above

distance lower bounds the geodesic distance:

$$d_{\mathcal{G}}(C_i, C_j) \geq d_{\mathcal{LE}}(C_i, C_j). \tag{16}$$

In practice, we observe that the Log-Euclidean distances closely approximate corresponding geodesic distances for region covariance descriptors. Hence, we use Log-Euclidean distances as a proxy for near neighbor search. As we show in the following experiments, this approximation still provides meaningful results in terms of finding near neighbors of covariance matrices in the geodesic distance sense.

Since log-euclidean distance involves the Frobenius norm of the matrix, we can perform the equivalent computation by vectorizing the corresponding matrix and then computing its L_2 norm instead.

$$L = \| \log(C) \|_F = \|\mathbf{c}\|_2. \tag{17}$$

Thus given vectors $\mathbf{c_i} \equiv \log(C_i)$ and $\mathbf{c_j} \equiv \log(C_j)$, the geodesic distance between C_i and C_j can be approximated by the Euclidean distance as:

$$d_{\mathcal{G}}(C_i, C_j) \approx \|\mathbf{c_i} - \mathbf{c_j}\|_2. \tag{18}$$

Since we can obtain the corresponding vector representations for each region covariance descriptor, we can now use Euclidean space LSH as in [Datar et al., 2004] for performing locality sensitive hashing that respects distances on the Riemmanian manifold.

In the preprocessing step, we perform the conversion to vector space of each covariance matrix after taking its logarithm. The vectors thus obtained are hashed using the standard Euclidean LSH described above. Fast sublinear time nearest neighbor search can then be performed.

6.6 Experiments with region covariances

In this section, we experimentally demonstrate the speedups achieved via LSH on region covariance descriptors, while respecting their geodesic distance. The covariance

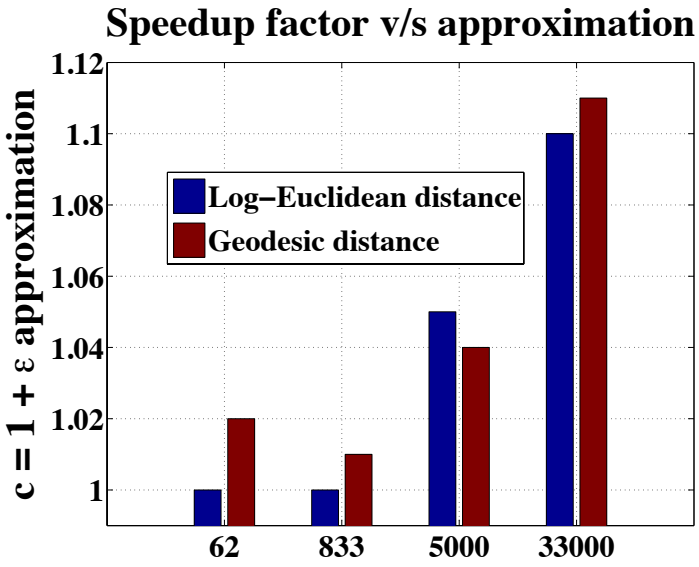


Figure 23: Speedups achieved with their corresponding approximation ratios (achieved nearest neighbor distance / true smallest distance) using the Log-Euclidean and Geodesic distance metrics. Note that the algorithm itself uses Log-Euclidean approximation, however, as the figure shows, Geodesic distance follows a very similar pattern. The speedups achieved are very large, for instance a 33000-fold speedup (4 orders of magnitude) with only a 10% (1.1) approximation to the closest neighbor. This allows us to scale to datasets with up to a million images even with covariance descriptors.

matrix is regularized by adding a diagonal scaled identity matrix in order to avoid numerical issues when the matrix is close to being rank-deficient. Small values of the order of about 10^{-8} seem to work well for the experiments.

The data consists of about 100000 region covariance descriptors (each as a 8 x 8 matrix) corresponding to various image patches. The data is obtained from the Microsoft Kinect sensor that allows depth estimation and comprehensive point cloud matching. It consists of 8 classes of objects, each having multiple images in the database. The data extraction process and other details are mentioned in [Fehr et al., 2011]. The task is, given a descriptor, to find its nearest neighbor for image matching in order to do object recognition using a k-NN classifier. Thus, fast near neighbor retrieval is immensely important for speedy real-time recognition. Figure 23 shows the speedup factors achieved along with their approximation ratios with both the distance metrics: Log-Euclidean and Geodesic. Unlike vectorization which destroys the manifold structure of the data [Sivalingam et al., 2009], the proposed method respects it, while still providing vector-space approximate search algorithms through the use of the Log-Euclidean approximation.

Further, we also performed experiments with a database of 1 million covariance matrix descriptors. In this case, linear scan was too slow to run experiments with. On the other hand, with the above technique we were able to obtain approximate nearest neighbors for each query in about 0.2 seconds on average, with unoptimized Python code. The result shows that potentially real-time detection and recognition with complex descriptors is possible without compromising retrieval accuracy.

7 Query Synthesis + Selective Sampling

As mentioned in Chapter 3, active selection has been approached via two methods: query construction or synthesis, and selective sampling. In the former, a query is constructed given the current labeled samples (and possibly, the current model), so as to maximize information gain or any other measure that helps improve future performance. Our focus in this thesis has been on selective sampling, in which informative queries are sampled from a pool of unlabeled examples.

Query construction can be problematic when the queries thus generated are meaningless to the human labeler or do not otherwise make sense in the problem domain. This observation was first made by Lang and Baum [1992] when they tried to use membership query synthesis for handwritten digit recognition. They observed that synthesized queries consisted of shapes which could not be reasonably classified as belonging to any digit which was problematic for the human labelers.

Another problem with query synthesis occurs when classification is performed by extracting features from data samples. In such a scenario, synthesizing a query implies deriving a feature descriptor, from which it might be impossible to obtain a data sample that can be queried to a human. For instance, consider the example of object detection in images. Typical features used are histograms of oriented gradients [Dalal and Triggs, 2005] which are extracted from the images by computing the gradient directions in the image and binning them appropriately. Given a model and training data in the histogram domain, suppose we synthesize a new query (a histogram) and seek its associated label. It is not possible to generate an image which corresponds to the given histogram (since the mapping is one-to-many, unique reconstruction is theoretically impossible, but even for one-to-one mappings, performing the inversion might be computationally intractable). For the task of object detection, a histogram query is meaningless to the human. Thus, query synthesis can be problematic when a feature extraction process is used to identify important cues that aid the classification

process.

On the other hand, query synthesis has the advantage that query time is independent of the unlabeled pool size. As data sizes grow, this speed advantage is of primary importance in scaling the learning process.

In this chapter, we demonstrate a new formulation of active learning by combining query synthesis with uncertainty sampling. The system exploits the speed advantage of query synthesis while at the same time enforces the queries to be sampled from the pool ensuring that they represent meaningful data samples in the context of the problem. The main idea is to synthesize informative queries given the current labeled data, and efficiently find nearest neighbors in the unlabeled pool using Locality-Sensitive hashing techniques described in the previous chapter. In order to ensure that informative samples are queried, density of the query region can be taken into account, or simply the distance between the constructed query and its nearest neighbor in the data. We employ variants of locality-sensitive hashing algorithms that provide sublinear query time (w.r.t. unlabeled pool size) and thus ensure fast retrieval of near neighbors allowing efficient active learning for very large datasets. We perform experiments with pool sizes of up to a million images on a standard laptop computer, where each active learning iteration requires only about 1 second for querying.

7.1 Membership query synthesis

Given training data and an associated classification model, it is intuitive to construct data samples that are potentially informative once the labels are given. Similar to pool-sampling, query synthesis can use various notions of informativeness, some of which might be more suitable for certain applications: information theoretic measures like entropy, information gain, or uncertainty measures, version space reduction, etc.

Motivations for synthesizing queries comes from very early work in active learning and artificial intelligence [Angluin, 1988; MacKay, 1992]. Angluin [2001] shows that for finite problem domains, efficient querying is possible. In the regression domain, Cohn et al. [1996] demonstrate query synthesis for the application of predicting robotic hand coordinates.

A very interesting application of query synthesis has been demonstrated by King

et al. [2004, 2009]; they let a robotic system ('robot scientist') perform a series of experiments to identify metabolic pathways in yeast. The key importance of active learning is to minimize the number of experiments to get a better understanding of the pathways.

On the other hand, as mentioned before, Lang and Baum [1992] came across an unexpected problem when synthesized queries were meaningless in the domain of the problem. Their particular experiment had digit images which did not appear like any of the 10 digits, and therefore were not appropriate queries. Similar problems can be expected for other areas such as text classification, music classification into genres, etc.

From these partial successes, it is reasonable to believe that query synthesis may be very useful where the space of queries is not restricted in terms of physical interpretations. However, for our application of image classification, the space of meaningful image descriptors is extremely limited – as such, a sampling based method for querying images from an unlabeled pool is particularly suitable.

7.2 Query construction and nearest neighbor search

We propose a very simple way to combine the benefits of both approaches. One can synthesize a membership query from the data, and then find nearest neighbors from the pool. If there are no neighbors 'close enough' to the synthesized query, a new query can be generated. In other words, data density at the query location can be used to guide the search in regions of higher density, where a near neighbor is a close approximation to the original query. The idea is illustrated in Figure 24.

One can formulate various query synthesis mechanisms depending on the classifier. For instance, with k-NN classifiers, one idea is to look at a point that lies midway between two clusters (say on the line connecting the cluster means), each cluster representing one class or category. The distance can be measured from the cluster means, or the average distance to each point in the cluster can be considered instead.

In our first approach, the queries are synthesized so as to lie at the midpoint on the line joining two cluster centers. First, cluster means are computed by finding the average of the data points belonging to each class. Suppose that training data

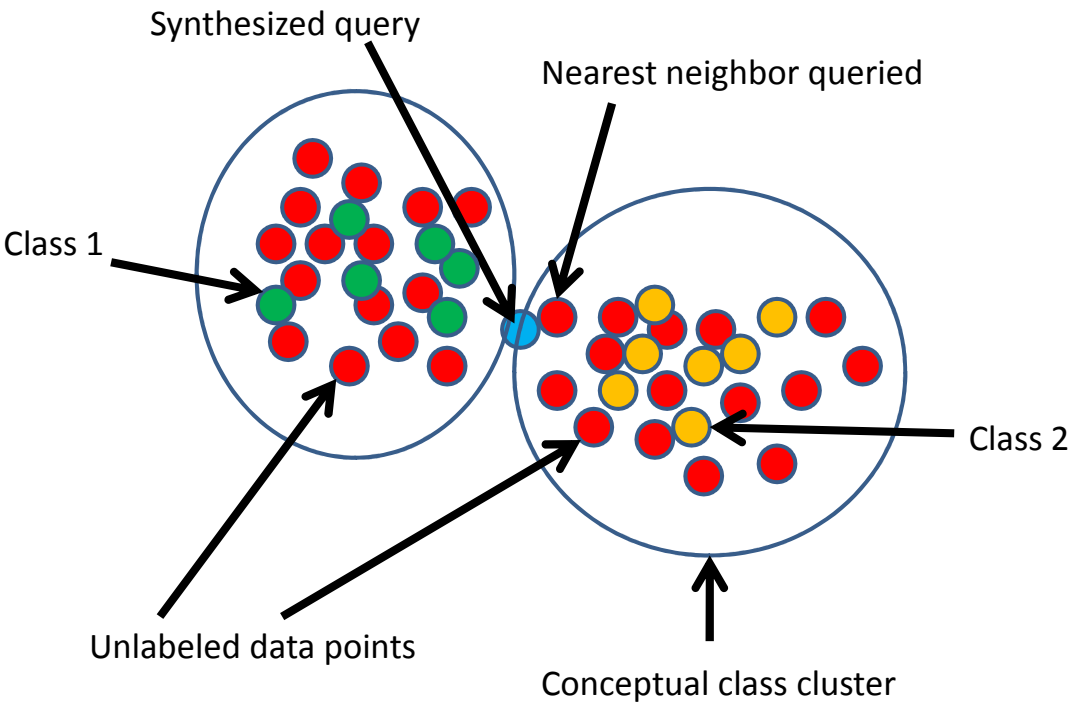


Figure 24: Illustration of the query synthesis. The training data from two classes form two conceptual clusters which can be considered the current model. A new query is synthesized using some measure, say, uncertainty of class membership. In the abvoe example, a sample that is most confusing given the current cluster model is synthesized for query. The red dots indicate unlabeled data. The system then finds nearest neighbors from the unlabeled pool and uses that as the query. Typically, a maximum distance threshold is used so that the samples queried are representative of the synthesized queries.

consists of points $x_1, \ldots, x_n$, for each $x_i \in \mathbb{R}^d$. The mean of cluster k (consisting of all data points belonging to class C_k) can be computed as

$$m_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i. \tag{19}$$

The query is then constructed as the mean of two clusters. Given clusters C_i and C_j, the synthesized query is

$$q_{ij} = \frac{m_i + m_j}{2}. \tag{20}$$

Note that the weights can be adjusted to consider density and cluster compactness to lie closer to one of the clusters instead of their mean. Here, we simply use the mean to illustrate the idea of query synthesis. More sophisticated mechanisms that use domain knowledge can easily be applied.

Intuitively, the synthesized query lies in the 'uncertainty' region between two classes, information about which can improve future performance. To implement the idea, cluster representatives are maintained and updated as and when new training samples are added to the data. Given the representatives, queries can be synthesized in constant time with respect to the data size.

Once the query is synthesized, we first seek its k nearest neighbors in the unlabeled data. Once this subset has been identified, pool-based selective sampling can be employed to finally chose the queries amongst this subset that are most informative. The above formulation ignores density of the data at the point q_{ij}. Consequently, nearest neighbors to the synthesized query might be far away, thus not being representative of the synthesized query. One way to overcome the problem is to synthesize many queries with varying parameters, and then chose nearest neighbors to them for query, only when they are within a certain distance from the synthesized query.

One the other hand, one can also use the pool-based selection measures used to decide whether to query certain samples or not based on the scores obtained on them.

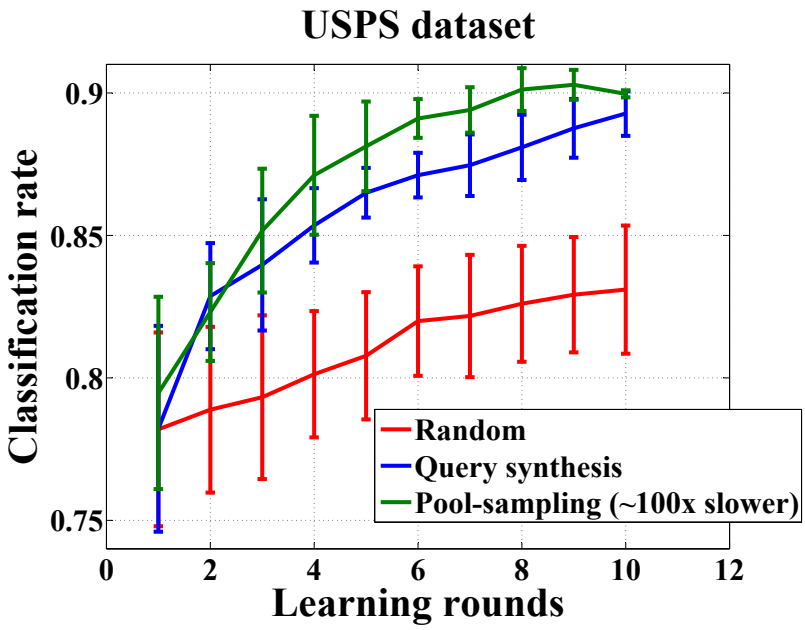


Figure 25: Illustrative example showing that synthesized queries perform only slightly worse than comprehensive uncertainty sampling on the entire pool, while giving two orders of magnitude speedup. This approach is particularly useful when informative queries can be synthesized given domain knowledge. We note that using seed samples from training data to initiate a near neighbor search is often more useful, especially when synthesis is not trivial, but speedups are desired.

7.3 Illustrative experiment

We illustrate the performance of the above method by an experiment on USPS dataset of handwritten digits from the UCI repository [Asuncion and Newman, 2007], and a nearest neighbor classifier. For this illustrative experiment, the data was divided into 2 classes, each containing 5 digits. The uncertainty score is defined as the ratio of the distances of a sample point to its nearest (labeled) neighbor to the distance to its second nearest neighbor. The closer the ratio to 1, the more uncertain the sample, and thus forms a viable selection candidate. Three approaches are compared: (i) standard selective sampling with the distance ratio measure on the unlabeled pool, (ii) the combination of synthesized queries with approximate nearest neighbor search, along with a final selection step using the uncertainty measure as in (i), and (iii) random selection. Figure 25 shows the results. Observe that the proposed method is only slightly worse that selective sampling, however, due to the fast approximate nearest neighbor search, provides about a 100-fold speedup. This result is extremely promising when unlabeled pools are large, and strong notions of query sysnthesis are available through domain knowledge.

7.4 Query synthesis vs. seed samples for search

As mentioned before, query synthesis is an appealing way for gathering information in many domains. However, it might not always be feasible for classification problems, wherein no notions of informativeness are available or the corresponding synthesis formulations are computationally hard. In such scenarios, seeding the search by using samples in the training set performs better. Another disadvantage of synthesis is that it can only be done in finite domains with explicit feature representations. In kernel-classifiers such as SVM, the explicit feature representation of data points may not be available (or even finite). In such cases, starting with seed samples (as was done in the previous Chapter) can be thought of as a proxy to syntehsizing the query. Furthermore, the search need not be restricted to one iteration. Searching for nearest neighbors and then choses the most informative ones from those (according to the active selection criterion) can be done iteratively until subsequent searches do not yield further gains. In our experiments, we did not find any benefit using such an

extension, but it can be potentially useful for extremely sparse training data which has a specific sampling bias, that leads to training samples that are away from the desired classifier boundary.

8 Batch-Mode Active Learning

Our goal is to focus on active selection in large, multi-class problems that are typical in real-world classification scenarios. Typically, the process of active example selection has been iterative – the classifier queries for labels on certain examples which the human provides, followed by a step of classifier retraining. Such interaction is problematic on two fronts: i) training a classifier at each iteration or round poses computational challenges, especially for classifiers that cannot be trained incrementally; and ii) since the human has to input new labels at each round, the process can be cumbersome, indicating interactive inefficiency.

Most work on active learning in binary problems [Freund et al., 1997; Campbell et al., 2000; Balcan et al., 2007; Tong and Koller, 2001; Tong and Chang, 2001; Kapoor et al., 2007b] as well as in multi-class classification [Joshi et al., 2009; Kapoor et al., 2007a; Qi et al., 2008; Jain and Kapoor, 2009] has focused on single return or iterative active learning. In [Vijayanarasimhan and Grauman, 2008], the authors employ user inputs at multiple levels of granularity, also referred to as multi-level annotations. Holub et al. [2008] propose entropy-based active learning that can handle batch-mode selection in principle, however, the approach is prohibitively expensive in practice. Recently a few researchers have proposed batch-mode selection algorithms [Brinker, 2003; Hoi et al., 2006], however these are restricted to only binary classification. Efficient batch-mode selection is vitally important in large multi-class classification problems in order to make the methods practically appealing and computationally feasible.

The chapter is organized as follows. We first discuss similarities between batch-mode active learning and experiment design and optimal sensing problems. There are some key differences which make iterative active learning beneficial compared to batch-mode selection which are also discussed. The chapter then outlines different formulations of the problem, and describes greedy algorithms for batch selection,

along with approximation guarantees where applicable. Note that most of the computational problems for batch-mode learning are NP-hard, and it is crucial to develop approximation schemes and heuristics.

8.1 Optimized information gathering

Batch-mode active learning is intimately related to work on experiment design [Federov, 1972], where the objective is to formulate an optimal measurement strategy to gather information about variable(s) of interest. More recently, there has been renewed interest in optimized sensor placement [Krause, 2008] so as to cover physical or conceptual regions of interest in order to gain information about interesting physical phenomenon such as temperature changes, water quality, outbreaks in networks, abnormal activity in camera streams, etc. In this section, we review the related work in the area of optimized sensing, and mention relevant performance bounds.

The main goal in experiment design and optimized sensing problems is (paraphrased from [Krause, 2008]) to know which observations to make in order to gather the most useful information cost-effectively. Krause [2008] provides a comprehensive analysis of the different formulations of such design problems, corresponding hardness results, approximation algorithms, and similarities between *a-priori* and sequential designs.

8.2 Quality of sensing

For review, we follow the discussion by Krause [2008]. Assume a utility function $u(x_v, \mathcal{S})$ where x_v is the current state of the world, and $\mathcal{S} \in \mathcal{A}$ is the set of observations chosen. The sensing quality of $\mathcal{S}$ is then given by its expected utility:

$$U(\mathcal{S}) = \int p(x_v)u(x_v, \mathcal{S})d_{x_v}. \tag{21}$$

The utility function maps a given set of observations into a real number quantifying the sensing quality of that set. The goal is then to chose a set of observations that maximizes the sensing quality. Note that the implicit assumption above is that there exists a known prior distribution capturing the state of the world. The distribution

is hard to estimate in practice, both from an accuracy perspective and in terms of computational requirements. Most work thus makes certain simplifying assumptions about the distribution in order to maximize the expected utility.

Consider the utility function [Lindley, 1956]:

$$u(x_v, \mathcal{S}) = \mathcal{H}(\theta) - \mathcal{H}(\theta|x_{\mathcal{S}}). \tag{22}$$

Maximizing the expected utility corresponding to the assumption that the most informative observation set is the one that reduces entropy the most.

Many variations of the utility function have been used in the literature for modeling sensing problems in different domains, and lead to various design strategies. For example, the entropy of a set of variables can be used as a utility function. Note that this differs from the above, since the entropy criterion choses observations that are *most uncertain*, whereas entropy reduction choses observations that lead to *maximum reduction in entropy after the observations are made.*

Another popular measure of utility is based on the mutual information between sets of observations. Specifically,

$$U(\mathcal{S}) = \mathcal{H}(\mathcal{X}_{\mathcal{B}}) - \mathcal{H}(\mathcal{X}_{\mathcal{B}}|\mathcal{X}_{\mathcal{S}}) = MI(\mathcal{X}_{\mathcal{B}}; \mathcal{X}_{\mathcal{S}}), \tag{23}$$

where MI stands for mutual information. In such a scheme, the goal is to select new observations that *provide the most information* about a set of samples of interest $\mathcal{B}$. The literature in this area refers to the mutual information criterion as Bayesian *D-optimality* [Chaloner and Verdinelli, 1995].

Another approach to observation selection is to minimize the predictive variance of the system (i.e., minimizing the variance of predictions made by the model on unknown locations). This criterion corresponds to Bayesian *A-optimality*. Please see [Krause, 2008] for details.

As mentioned before, all of the above formulations require estimation of the probability distribution for the current state. This estimate is hard to obtain with limited data samples which can hurt performance in practice.

Furthermore, note that the above formulations seek to maximize information gained through a set of observations. In contrast, in the active learning task for

classification, the goal is to improve future classification, which may not correlate with the information gain. We would like to draw intuitive analogies with the difference between classification and density estimation – simple classifiers such as Naive Bayes are poor density estimators; this limitations often does not hinder their classification performance, even in the presence of fairly strong conditional independence assumptions. Intuitively, gathering more observations is not necessarily useful in the active learning task if those observations do not aid in the *discrimination objective* itself.

Our experiments also show that the $A-optimality$ and $D-optimality$ criteria often fail to improve upon random selection of observations. We note that this is the primary difference between experiment design applications that seek to maximize the information gain, and active learning, where the task is more focused on discrimination of data samples into a set of categories.

Another class of methods use decision-theoretic measures for observation selection. The goal is to chose observations so as to reduce the risk in making certain decisions, while the cost of observations can also be incorporated. These methods typically define a particular *Value of Information* [Krause and Guestrin, 2005] criterion which is then maximized. The decision theoretic formulation is more suitable to the task of active learning, as was also discussed in Chapter 5.

In the following, we discuss our proposed methods for multi-class batch-mode active learning.

8.3 Challenges in multi-class batch-mode selection

Actively selecting a single example for human labeling (single-return) requires a selection measure for querying *useful* examples at each iteration. By appending the newly labeled example to the training set, a new classifier can be trained for the next iteration. The active selection measure needs to be computed at every iteration since it depends on the current trained classifier. This implicitly minimizes redundancy in the queried examples, since if the classifier is confident on certain examples, they are not queried on future rounds. In batch-mode selection, the redundancy between examples needs to be accounted for explicitly. The primary challenges therefore are

the following:

- Along with a measure of 'usefulness' of examples for active selection, we need a criterion to evaluate redundancy of examples. Finding a measure for example redundancy is especially hard in multi-class problems, since redundancy depends heavily on the classifiers employed, the feature space, and class populations among others. It is thus not straightforward to generalize measures of example redundancy from binary to multi-class classification.

- Even if we have redundancy measures, batch-mode selection poses a big computational bottleneck. Consider that we need to select a batch of size k from an unlabeled data pool of size n. The number of possible batches that can be selected is ${}^{n}C_{k}$. n and k are typically large – we therefore run into intractable subset selection problems.

Figure 26 shows the block diagram of the batch-mode learning setup employed here. Note the absence of the feedback loop (indicating an iterative algorithm). The sample selection is one-shot.

In the following, we demonstrate the motivation for capturing redundancy in the samples. Consider a naive batch-mode selection method that simply chooses the best k samples according to their individual uncertainty scores. As a consequence of ignoring redundancy between samples when choosing batches, iterative (or single return active learning) significantly outperforms batch-mode selection, as illustrated in Figure 27.

8.4 Capturing redundancy

In terms of information gathering and optimal sensing, redundancy is often considered to be the amount of mutual information between two data samples. Measures such as conditional information gain, etc. are used to capture the new information that can be gained. Since in the active learning domain, we are interested in redundancy *with respect to the current model*, we work with class membership probability estimates that are obtained from the current classifier. As mentioned in previous chapters,

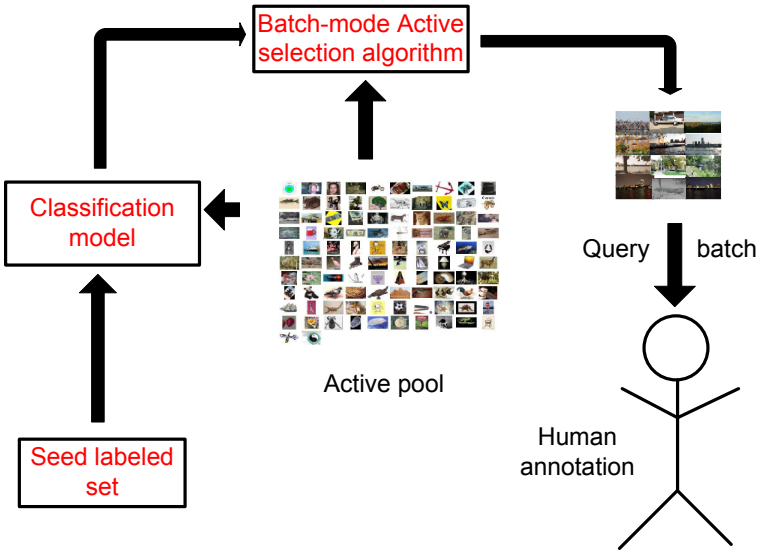


Figure 26: Batch-mode selection model proposed in this chapter. Note the absence of the feedback loop in batch-mode selection – this avoids multiple retraining of the classifier and provides easier user interaction. However, batch selection needs to explicitly handle example redundancy while being computationally tractable – the proposed methods address these problems.

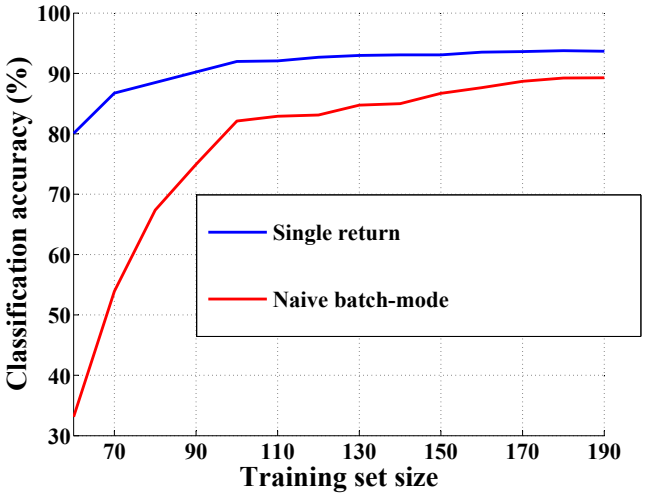


Figure 27: Difference between iterative (single-return) and batch-mode active learning. A large reduction in classification accuracy occurs due to naively selecting images in batch without considering redundancy.

such estimation can be performed by fitting appropriate functions to margins for margin-based classifiers, and are directly available for probabilistic models.

Jensen-Shannon divergence is a popular variant of the Kullback-Leibler divergence [Kullback and Leibler, 1951] and is used for estimating differences between probability distributions. Given two distributions P and Q, the Jensen-Shannon divergence (JS) is given by:

$$JS(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M), \tag{24}$$

where $M = (P+Q)/2$, and $KL(P\|Q)$ denotes the Kullback-Leibler (KL) divergence between distributions P and Q. The JS divergence can be viewed as a symmetrized and smoothed version of the corresponding KL divergence. From a theoretical perspective, the Jensen-Shannon divergence quantifies the reliability with which a decision can be made regarding the sample, as to whether it comes from the joint distribution of two variables or their product distribution, given that it comes from one of those two.

Some advantages of the JS divergence are that it is always finite, and its square root is a metric, which can be useful in some applications. Generalizations of the above to multiple variables is also possible, even with non-uniform weights like so:

$$JS(P_1, \ldots, P_n) = \mathcal{H}(\sum_{i=1}^{n} w_i P_i) - \sum_{i=1}^{n} w_i \mathcal{H}(P_i), \tag{25}$$

where w_i is the weight for the distribution P_i, and $\mathcal{H}(\cdot)$ denotes the entropy function.

Given a set of samples and their corresponding class membership distribution, the JS divergence can be used to capture their similarities, which takes into account the current classifier model. Thus JS divergence can then be used to select a set of samples so as to minimize redundancy of the batch. However, the subset selection problem still remains. In order to overcome that, we apply a greedy iterative algorithm that selects the next sample that maximizes the JS divergence, subject to the constraint that it is informative. In practice many ways can be used to accomplish this. For example, we can choose a few most informative samples, and then select the one that maximizes diversity. On the other hand, we can also chose the samples that maximize diversity and then pick the most informative amongst those. Experiments show that the former approach gives better results, so we demonstrate results using that approach.

The first sample selected is the one that has the maximum uncertainty score (similarly, any other measure could be used). Next, the top n samples that have the highest uncertainty score are chosen from the set. The sample that leads to the highest diversity *of the chosen set so far along with the new sample* measured by their JS divergence is then picked. The iterative process continues until the number of chosen samples equals the desired batch size. Labels are requested on the chosen samples and the classifier is retrained. Figure 28 describes the greedy algorithm using JS-divergence as the diversity measure. The greedy algorithm does not have any performance guarantees (such as approximation bounds relative to the optimal solution), however, in practice we observe very small differences between exhaustive subset search and the greedy algorithm for small datasets. For larger data sizes, the exhaustive search is too slow to gather such data.

Note that Melville [2003] also use JS divergence to capture diversity in an active

Input: Unlabeled data pool $\mathcal{A}$, batch-size k

```
1.  S := {φ}, the current batch of examples;
2.  for i := 1 to k, do
3.      foreach element x ∈ A\S, do
4.          Compute its uncertainty score U(x);
5.      Choose n elements having the highest uncertainty, denoted by set U_n;
6.      if S = {φ}
7.          then choose sample x* that maximizes U(x);
8.          else
9.              foreach element x ∈ U_n, do
10.                 Compute the JS-divergence of set S ∪ {x}, denoted by JS(x);
11.             Choose the sample x* that maximizes JS(x);
12.             Update batch, S := S ∪ {x*};
13. return S.
```

Output: The actively selected batch $\mathcal{S}, |\mathcal{S}| = k$.

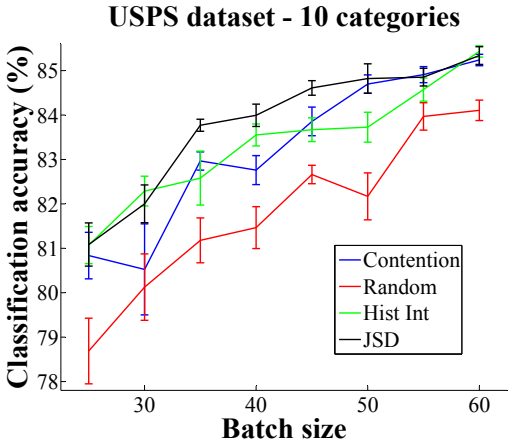
Figure 28: A greedy batch-mode selection algorithm using Jensen-Shannon divergence as the set diversity measure. The selection is biased towards informative samples which are diverse at the same time.

learning setting, however, their approach differs in the fact that they use JS divergence measure to generate a set of *diverse hypothesis* for active learning. On the other hand, we use JS-divergence for directly capturing diversity of the data samples queried for labels.

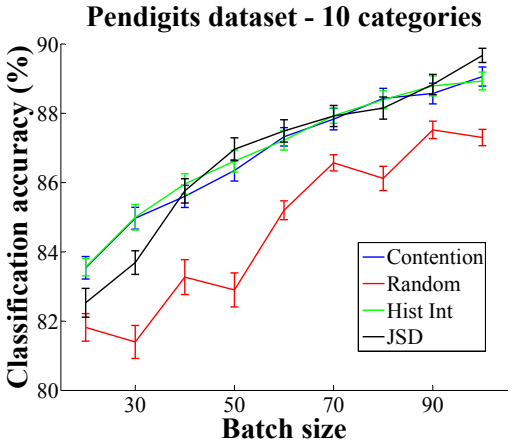
The above algorithm picks samples that are informative, while at the same time reasonably diverse so that redundant training samples are avoided. At each iteration of the above algorithm, most uncertain samples are chosen, and the JS divergence is computed after appending them to the current batch. The idea of this greedy method is similar to work on feature selection, where greedy algorithms are typically used to avoid the associated computational intractability, while also ensuring diversity in the set of selected features using mutual information type measures between features. For

example, Peng et al. [2005] propose a minimum-redundancy, maximum-relevance criteria for feature selection. The minimum redundancy aspect ensures diverse features to be selected, while the maximum relevance criterion forces the individual features to be highly informative about the target variable.

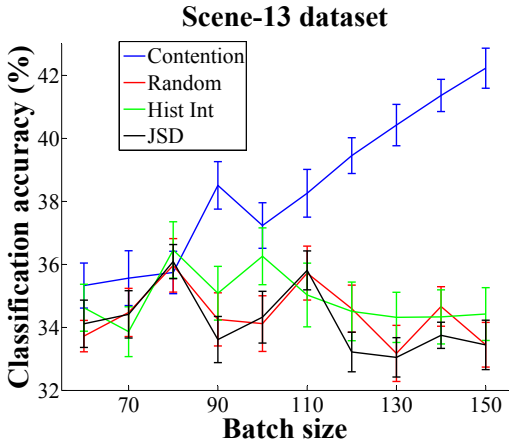
Analogously in our case, the uncertainty measure (used as a proxy for informativeness of a sample) ensures that the chosen sample is useful from the standpoint of obtaining new information for the classification model. On the other hand, the redundancy is captured by the JS divergence between different features.



(a)



(b)



(c)

Figure 29: Batch-mode active selection v/s random batch selection on different datasets: (a) USPS, (b) Pendigits, (c) Scene-13. Redundancy measures used – Contention: Classifiers in contention, Hist Int: Histogram intersection, JSD: Jensen-Shannon divergence. Similar results observed on other datasets also.

In the experiments, we compare the approach of using the JS-divergence with random selection, and other approaches used in the literature for evaluating similarities between two probability distributions: i) histogram intersections, and ii) classifiers is contention; both are described below.

Histogram intersection is a measure of similarity between two histograms or discrete probability densities. For distributions P and Q, it is defined as

$$\mathbf{I}(P, Q) = \sum_i \min(P_i, Q_i). \tag{26}$$

Intuitively, the intersection captures the 'overlap' between two histograms, and thus their similarity. Thus, lower the intersection value, higher is the diversity between samples. Unfortunately, extending this measure to the multi-class case does not provide good results (possibly due to the number of classes in the data). Instead, we compute the score for a set as the mean of the intersection scores for all pairwise samples. This captures one intuitive notion of 'average' similarity between histograms,

and works reasonably well in practice.

For multi-class problems, a concept referred to as "classifiers in contention" was introduced in Chapter 4. Refer back to the estimated distribution in Figure 3. From the distribution, we can see that 'Class 4' is the most likely category of the example. Since we use one-vs-one classification, all the classifiers that separate class 4 from the other classes are the classifiers in contention – in particular, the classifier separating classes 4 and 5 is the most likely classifier in contention. This concept can also be employed for forming an interference measure – if two examples are likely to affect two different classifiers, they likely carry different information and are not redundant. Through this idea, we can capture the potential redundancies in *multi-class* problems, which is much more challenging that redundancy estimation in binary classification. Similarly to histogram intersection, we generalize this score to the set of samples by taking the average of the pairwise scores for all samples in the batch.

Figure 29 shows results using the above measures on different datasets. We can see that on the USPS and Pendigits datasets, all the redundancy measures outperform random selection significantly, showing that even in multi-class batch-mode selection, a lot of annotation effort can be reduced by the proposed algorithm. On the Scene-13 dataset, JSD and Histogram intersection perform poorly, giving accuracy values similar to random example selection. However, the method using classifiers in contention beats all other methods by a large margin. The result indicates that capturing redundancy is crucial to good performance. In this case, the 'contention' method looks greedily for distributions that peak at the same categories, while 'JSD' and 'histogram intersection' look at the entire distributions, and therefore fail to capture the corresponding example redundancies. The reason is perhaps similar to why a multi-class margin approach beats entropy-based selection: the margin approach looks only at the most likely classes, thereby capturing the most essential aspects of the classification objective.

8.5 Coverage formulations

Previously, a batch-mode active learning approach that uses redundancy measures to select informative and diverse sets of samples was described. In this section,

we look at batch-mode learning from the perspective of maximizing *coverage of the chosen samples* so that they are representative of the data points in the unlabeled set. The flavor of coverage formulations is similar to spatial sensing problems, and also much studied facility location problems in combinatorial optimization. Unlike facility location objectives however, costs are not associated with setting up particular facility locations (choosing certain samples for query), however, a budget of the total number of queries is placed.

The following sections describe two approaches, one using greedy optimization of a submodular objective function, and another using farthest first heuristic for finding diverse samples.

8.6 Batch-mode learning with nearest neighbor classifiers

In a nearest neighbor (NN) classifier, for each data point, the label is obtained by finding the label of the nearest labeled point. Our goal here is to maximize the coverage of the training points. To this end, we want to ensure that all points have their nearest labeled point at a bounded distance. Consider that the distance between two points is given by the function $\mathbf{d} \in \mathbb{R}^d \mathrm{x} \mathbb{R}^d \rightarrow \mathbb{R}$. Denote the training set (to be chosen by active selection) as $\mathcal{S}$. Say the budget allows for only a certain number of labeled samples to be collected, such that $|\mathcal{S}| \leq n$. Also, denote the active pool from which samples are to be chosen by $\mathcal{A}$.

The minimum distance for any point x from the training set $\mathcal{S}$ is given by:

$$\min_{i:x_i \in \mathcal{S}} \quad \mathbf{d}(x_i, x), \tag{27}$$

The maximum distance for any point in the dataset to its nearest neighbor is then:

$$\max_{j:x_j \in \mathcal{A}} \quad \min_{i:x_i \in \mathcal{S}} \quad \mathbf{d}(x_i, x_j). \tag{28}$$

Our objective, as mentioned previously is to minimize the above distance, so that for each data point, there exist at least one training point at a bounded distance. Hence

the goal is to solve the following

$$\mathcal{S} = \underset{|\mathcal{S}|\leq n}{\operatorname{argmin}} \max_{j:x_j\in\mathcal{A}} \min_{i:x_i\in\mathcal{S}} \mathbf{d}(x_i, x_j). \tag{29}$$

For easier analysis, we instead define the objective function to be *maximized* as:

$$\begin{aligned} \mathbf{Q}(\mathcal{S}) &= \mathbf{d}_m - \max_{j:x_j\in\mathcal{A}} \min_{i:x_i\in\mathcal{S}} \mathbf{d}(x_i, x_j), \text{ for } \mathcal{S} \neq \phi \\ &= 0, \text{ otherwise.} \end{aligned} \tag{30}$$

where $\mathbf{d}_m \geq \mathbf{d}_{max}$, the maximum possible pairwise distance in the dataset. Thus, $\mathbf{d}_m$ is at least as large as the diameter of the data. As we will see later, the exact value of $\mathbf{d}_m$ is inconsequential for the actual implementation, since its value does not depend on the optimization variable.

Note that solving Equation (29) exactly is equivalent to maximizing $\mathbf{Q}$ given in Equation (30), subject to the budget constraint. Thus, we can define our problem $\mathbf{P}$ as:

$$\mathbf{P}: \max \ \mathbf{Q}(\mathcal{S}) \ \text{ s.t. } |\mathcal{S}| \leq n. \tag{31}$$

Even though the two optimization problems are equivalent when solved exactly, approximate solutions to the two problems are not equivalent, as we discuss below. However, we see that approximation algorithms still provide good empirical results in practice.

The above problem in Equation (29) is similar to the known k-center problem is combinatorial optimization, where the goal is to find k-cluster centers along with point assignments so as to minimize the maximum cluster radius. It can be shown that the problem is intractable by a reduction to SET-COVER, for instance.

However, exploiting the properties of the objective function, strong approximation guarantees can be provided. We explore two such approaches in this work:

- The above problem given in Equation (31) confirms to the paradigm of submodular optimization [Nemhauser et al., 1978], and thus greedy near-optimal

algorithms are applicable. A $(1-1/e)$ approximation guarantee can be provided for the corresponding maximization problem.

- It can be shown that a farthest-first heuristic algorithm for greedy selection of k centers leads to a near-optimal solution for Equation (29), with a 2-approximation for the minimization problem.

We perform empirical analysis of batch-mode active learning with both of the above approximation algorithms. Results show that the proposed methods improve the classifiers trained by active learning compared to random selection of sample batches. However, note that iterative active learning performs better than batch-mode selection, and should be used when iterative retraining and feedback are available. In principle, more information is available to the algorithm for choosing the next sample in an iterative setting – whereas in a batch-mode setup all decisions must happen *a-priori*. Consequently, more informative samples are chosen in iterative active learning settings. This is one reason why much work in optimal sensing and experiment design is not directly applicable to the active learning domain.

Claim 1 $\mathbf{Q}$ as defined in Equation (30) is a monotonically non-decreasing function.

Proof First note that for any $\mathcal{S} \neq \phi$, $\mathbf{Q} \geq 0$, since $\mathbf{d}_m$ is the at least as large as the largest pairwise distance in the point set. Further, it is easy to see that for any two sets $\mathcal{S}_1 \subseteq \mathcal{S}_2$, and $x_j \in \mathcal{A}$, $\min_{i:x_i \in \mathcal{S}_1} \mathbf{d}(x_i, x_j) \geq \min_{i:x_i \in \mathcal{S}_2} \mathbf{d}(x_i, x_j)$. Thus, the maximum such distance in the set $\mathcal{A}$ is also bounded, giving $\mathbf{Q}(\mathcal{S}_1) \leq \mathbf{Q}(\mathcal{S}_2)$. ∎

Claim 2 $\mathbf{Q}$ is a submodular set function.

Proof Assume $\mathcal{S}_1 \subseteq \mathcal{S}_2$ such that $\mathcal{S}_2 = \mathcal{S}_1 \cup \mathcal{S}$. Also, say x is a sample data point such that $x \in \mathcal{A}, x \notin \mathcal{S}_2$. Now

$$\mathbf{Q}(\mathcal{S}_2 \cup \{x\}) - \mathbf{Q}(\mathcal{S}_2) = \max_{j \in \mathcal{A}} \min_{i \in \mathcal{S}_2} \mathbf{d}(x_i, x_j) - \max_{j \in \mathcal{A}} \min_{i \in \mathcal{S}_2 \cup \{x\}} \mathbf{d}(x_i, x_j).$$
$$\mathbf{Q}(\mathcal{S}_1) - \mathbf{Q}(\mathcal{S}_1 \cup \{x\}) = \max_{j \in \mathcal{A}} \min_{i \in \mathcal{S}_1 \cup \{x\}} \mathbf{d}(x_i, x_j) - \max_{j \in \mathcal{A}} \min_{i \in \mathcal{S}_1} \mathbf{d}(x_i, x_j).$$

Denote

$$\begin{aligned} M_1(j) &= \min_{i \in \mathcal{S}_1} \mathbf{d}(x_i, x_j), \\ M_2(j) &= \min_{i \in \mathcal{S}_1 \cup \{x\}} \mathbf{d}(x_i, x_j), \\ \mathbf{Q}_d = \mathbf{Q}(\mathcal{S}_2 \cup \{x\}) - \mathbf{Q}(\mathcal{S}_2) &+ \mathbf{Q}(\mathcal{S}_1) - \mathbf{Q}(\mathcal{S}_1 \cup \{x\}). \end{aligned} \tag{32}$$

Thus,

$$\begin{aligned} \mathbf{Q}_d = \max_{jin\mathcal{A}} \min_{i \in \mathcal{S}_1 \cup \mathcal{S}} \mathbf{d}(x_i, x_j) - \max_{j \in \mathcal{A}} M_1(j) \\ + \max_{j \in \mathcal{A}} M_2(j) - \max_{j \in \mathcal{A}} \min_{i \in \mathcal{S}_1 \cup \mathcal{S} \cup \{x\}} \mathbf{d}(x_i, x_j). \\ = \max_{j \in \mathcal{A}} \min(M_1(j), \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j)) - \max_{j \in \mathcal{A}} M_1(j) \\ + \max_{j \in \mathcal{A}} M_2(j) - \max_{j \in \mathcal{A}} \min(M_2(j), \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j)). \end{aligned} \tag{33}$$

Denote $\delta = \max_j M_1(j) - \max_j M_2(j)$. Since $M_1(j) \geq M_2(j), \forall j$, we have $\delta \geq 0$, and

$$\begin{aligned} \mathbf{Q}_d = \max_{j \in \mathcal{A}} \min(M_1(j), \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j)) \\ - \max_{j \in \mathcal{A}} \min(M_2(j), \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j)) - \delta. \end{aligned} \tag{34}$$

We divide the analysis in 3 cases below.
Case 1:

$$M = \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j) \leq M_2(j).$$

Then,

$$\begin{aligned} \mathbf{Q}_d = \max_{j \in \mathcal{A}} M - \max_{j \in \mathcal{A}} M - \delta \\ = -\delta \leq 0. \end{aligned} \tag{35}$$

Case 2:

$$M = \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j) \geq M_1(j).$$

Then,

$$\begin{aligned} \mathbf{Q}_d &= \max_{j \in \mathcal{A}} M_1(j) - \max_{j \in \mathcal{A}} M_2(j) - \delta \\ &= 0. \end{aligned} \tag{36}$$

Case 3:

$$M_2(j) < M = \min_{i \in \mathcal{S}} \mathbf{d}(x_i, x_j) < M_1(j).$$

Then,

$$\mathbf{Q}_d = M - \max_{j \in \mathcal{A}} M_2(j) - \delta \leq 0. \tag{37}$$

From Equations (35), (36), and (37), we have $\mathbf{Q}_d \leq 0$. Thus from Eqn (32),

$$\mathbf{Q}(\mathcal{S}_2 \cup \{x\}) - \mathbf{Q}(\mathcal{S}_2) \leq \mathbf{Q}(\mathcal{S}_1 \cup \{x\}) - \mathbf{Q}(\mathcal{S}_1), \tag{38}$$

where $\mathcal{S}_1 \subseteq \mathcal{S}_2$, thus showing that $\mathbf{Q}$ is submodular. ∎

Intuitively, this means that adding an element to a smaller set presents more value than adding it to a larger set – the property of diminishing returns.

Given a submodular non-decreasing set function $\mathbf{Q}$ such that $\mathbf{Q}(\phi) = 0$, Nemhauser et al. [1978] show that a greedy algorithm gives an objective value no worse than a $(1 - 1/e)$ factor of the optimal. Note that even though the globally optimal solutions to both Equations (29) and (31) are the same, the approximation guarantees are not. Specifically, the greedy algorithm $(1 - 1/e)$-approximation bound does not hold for Equation (29), and the solution can be arbitrarily worse. To see this, assume that the optimal value $\mathbf{Q}^* = \mathbf{d}_m - d^*$, where d^* is the optimal distance of interest. Also, denote by $\hat{\mathbf{Q}}$ the achieved value of $\mathbf{Q}$, and $\hat{d}$ the corresponding distance. Let c be the

approximation factor. Then

$$\mathbf{d}_m - \hat{d} = c * (\mathbf{d}_m - d^*), \tag{39}$$

which implies that $\hat{d}$ can be far away from d^*, and the actual approximation obtained depends on $\mathbf{d}_m$, which is independent of our desired optimization variable. However, in our experiments, we often see approximations that are much closer to the optimal giving good results in practice.

Given near-optimality guarantees, tools in submodular optimization have been used extensively for problems in experiment design, sensor placements, network outbreak detections, etc. For example, in [Krause et al., 2008; Krause and Guestrin, 2005; Krause et al., 2011], submodular optimization techniques are used for effective sensor placement in Gaussian Processes and other graphical models.

8.7 k-NN and submodularity

Now we turn our attention to the k-NN classifier – unfortunately, the previous analysis does not apply here. It is straightforward to create a counter example showing that the analogous problem for k nearest neighbors does not give a submodular objective function. For example, consider a point x and a set $\mathcal{S}_2 = \{x_1, x_2, x_3, x_4\}, \mathcal{S}_1 = \{x_3, x_4\}$, so that $\mathcal{S}_1 \subset \mathcal{S}_2$. Further assume that the samples are at distances such that x_1 is the closest to x and x_4 is the farthest in $\mathcal{S}_2$. Also, let $\mathbf{d}(x, x_2) - \mathbf{d}(x, x_1) > \mathbf{d}(x, x_4) - \mathbf{d}(x, x_3)$. If we consider $k = 2$, and add a point $\hat{x}$ which is closest to x amongst all points, we can see that the property of diminishing returns is violated – i.e., set $\mathcal{S}_2$ encounters larger increase in objective value than set $\mathcal{S}_1$ according to Equation (30). Thus the corresponding objective function is not submodular.

Instead of looking at the maximum distance to the k^{th} nearest neighbor, one way might be to instead minimize the average distance. Note that this problem (also intractable in general) is slightly different from the k-median problem heavily studied in the literature. In the k-median problem, the goal is to minimize the sum of distances of points to their cluster centers.

Due to the computational intractability of the problem, we still use the 1-NN formulation even when using a k-NN classifier with k different from 1. This approach

Input: Unlabeled data pool $\mathcal{A}$, batch-size k

1. $\mathcal{S} := \{\phi\}$, the current batch of examples;
2. **for** $i := 1$ **to** k, **do**
3. **foreach element** $x \in \mathcal{A} \backslash \mathcal{S}$, **do**
4. Compute $\mathbf{Q}(\mathcal{S} \cup \{x\})$ using Equation (31);
5. Select the example $x^* = \text{argmax}_x \mathbf{Q}(\mathcal{S} \cup \{x\})$;
6. $\mathcal{S} := \mathcal{S} \cup \{x^*\}$;
7. **end**
8. return $\mathcal{S}$.

Output: The actively selected batch $\mathcal{S}, |\mathcal{S}| = k$.

Figure 30: A greedy batch-mode active selection algorithm.

performs well empirically.

Greedy algorithm

Figure 30 describes the greedy batch-mode active selection algorithm. As mentioned before, the algorithm achieves an objective function value that is within a $(1 - 1/e)$ factor of the optimal value for the optimization problem given in Equation (31). As mentioned before, the approximation guarantee does not give a useful bound on the optimization problem of Equation (29), however the empirical results are promising.

Computational requirements

Even though the above greedy algorithm is polynomial in the unlabeled data size and the batch size, it can still be slow in practice. Specifically, the algorithm has a time bound $\mathcal{O}(nk^2)$, with n points, and a batch size of k, and performs worse with larger batches.

Input: Unlabeled data pool $\mathcal{A}$, batch-size k

1. $\mathcal{S} := x_R$, $x_R \in \mathcal{A}$, a randomly chosen sample;
2. **for** $i := 2$ **to** k, **do**
3. Find $x^* \in \mathcal{A} \backslash \mathcal{S}$ such that $x^* = \text{argmax}_x \, d(x, \mathcal{S})$;
6. $\mathcal{S} := \mathcal{S} \cup \{x^*\}$;
7. **end**
8. return $\mathcal{S}$.

Output: The actively selected batch $\mathcal{S}, |\mathcal{S}| = k$.

Figure 31: Greedy farthest-first active selection algorithm.

8.8 Greedy algorithm for k-center

The optimization formulation given in Equation (29) is the k-center problem studied heavily in the literature. In this setting, greedy algorithms such as farthest first point selection have been explored for actively seeding clustering [Basu et al., 2002]. The algorithm begins with a point chosen randomly from the unlabeled pool, and at each iteration a new point is picked to be farthest from the current chosen set. The distance of a point to a set implies the distance from the point to its closest element in the set. The greedy farthest-first algorithm for the k-center problem is described in Figure 31.

Claim 3 The algorithm given in Figure 31 is a 2-approximate algorithm (provides a worst-case factor of 2 approximation) for the k-center problem of Equation (29).

Claim 4 Unless $P = NP$, there is no α-approximation algorithm from the k-center problem of Equation (29), for $\alpha < 2$.

See Theorems 2.3 and 2.4 in the book [Williamson and Shmoys, 2010] for com-

prehensive details on related approximation algorithms and the proofs of the above claims.

Even though the simplicity of the farthest-first algorithm is appealing, it tends to pick samples at the boundary of the data. As such, the chosen samples are not representatives of the data, and often lead to poor generalization. Since labeled samples influence the accuracy of classification of nearby points, choosing samples on the boundary (or outliers in a sense) does not improve classification. As we show in the next section, this problem can be alleviated by incorporating model information in the formulation.

8.9 Incorporating classifier information

The methods described so far confirm well to the coverage formulation, so that samples selected for training are distributed well across the training set. However, no information pertinent to classification is used – thus even though these methods successfully solve the optimization problem to near-optimality they do not work well in the classification setting. For instance, consider the case where the data is highly imbalanced, such that one class has many data samples, while there are other classes with very sparse populations. In this case, it is not useful to cover the data set with well-distributed training samples, since most such samples would be from the same class – instead, it would be more beneficial to actively seek sparser classes. Label information is thus extremely important in addition to coverage. In this section, we improve the models to account for current information from the training data, so that only data which is "informative" for the *current model* is chosen for training.

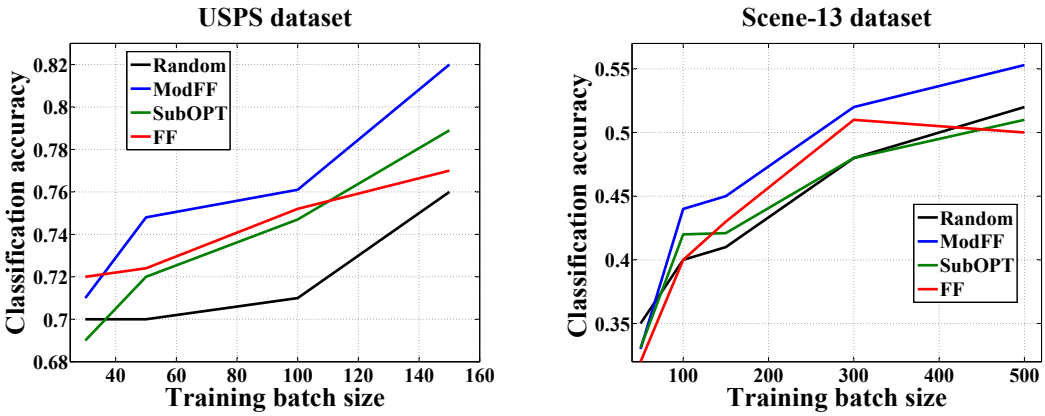
The main idea is to use model uncertainty to bias the method towards the selection of points for which there is classification uncertainty. Here we combine uncertainty sampling with coverage objectives in the previous sections to incorporate both pieces of information. The experiments demonstrate that this method significantly improves classification accuracy, and also requires lesser computation.

We require notions of uncertainty that capture the value of obtaining the label for a given data point. Also, we require the uncertainty measure to be applicable to classification problems with many classes, and be amenable to fast computation.

Here, we focus on the proportion of points coming from the different classes amongst the k nearest neighbors of a sample point. For instance if all k neighbors belong to the same class, the classification on the point is not uncertain, and choosing that sample to obtain the label might be redundant. On the other hand, if the k neighbors has each sample coming from a different class, then the classifier is uncertain about the membership of the sample point, and thus it is "informative" to query. More precisely, our uncertainty score is the difference between the number of points coming from the most populous class and the second most populous class amongst the k nearest neighbors. This confusion measure is similar to the notion of multi-class margin described previously [Cramer and Singer, 2001], and our active selection BvSB measure proposed in Chapter 4.

In order to incorporate this measure into the coverage formulation, we make a small modification to Equation 29. Instead of searching over the entire active pool for samples, we restrict the search to samples that give a high uncertainty score. Specifically, if the batch of examples to be selected is of size k, we simply choose $k' = mk$ most uncertain samples over which the more extensive optimization is carried out, for a multiplier m (say 5). The underlying hypothesis is that given informative samples, coverage is an important criterion to ensure good distribution of training samples. Apart from improved classification, we also get the benefit of substantially faster computation since the farthest first algorithms and greedy submodular set selection only need to be performed on the reduced informative sample set. Note that the most informative samples can be chosen in $\mathcal{O}(nk)$ time (since m is a constant), which is a factor of k improvement on the previous algorithms. The actual subset selection then runs in time independent of n, only relying on the required batch size k.

In this section we perform experiments on 3 different real-world datasets, namely US Postal service handwritten digits dataset (USPS) [Asuncion and Newman, 2007], the Letter recognition dataset (Letter) [Asuncion and Newman, 2007] and the Scene-13 dataset [Fei-Fei and Perona, 2005] consisting of images from 13 natural scene categories. For the USPS and Letter datasets, vectorized pixel values were used as features, whereas for Scene-13, 384-dimensional Gist descriptors [Oliva and Torralba, 2001] that give a scene summary were used.



(a) (b)

Figure 32: Classification accuracy values with increasing batch sizes for USPS and Scene-13 datasets. FF – farthest first greedy selection, SubOPT – greedy algorithm using submodular optimization, ModFF – Farthest first modified using informative sample subsampling. Note that SubOPT performs as bad as random for the Scene-13, perhaps due to very high dimensional data. Also, the improvements due to ModFF are smaller for Scene-13 than USPS. Standard deviation bars are not shown here to avoid clutter, however, note that the deviation values are extremely small and all the differences observed are statistically significant.

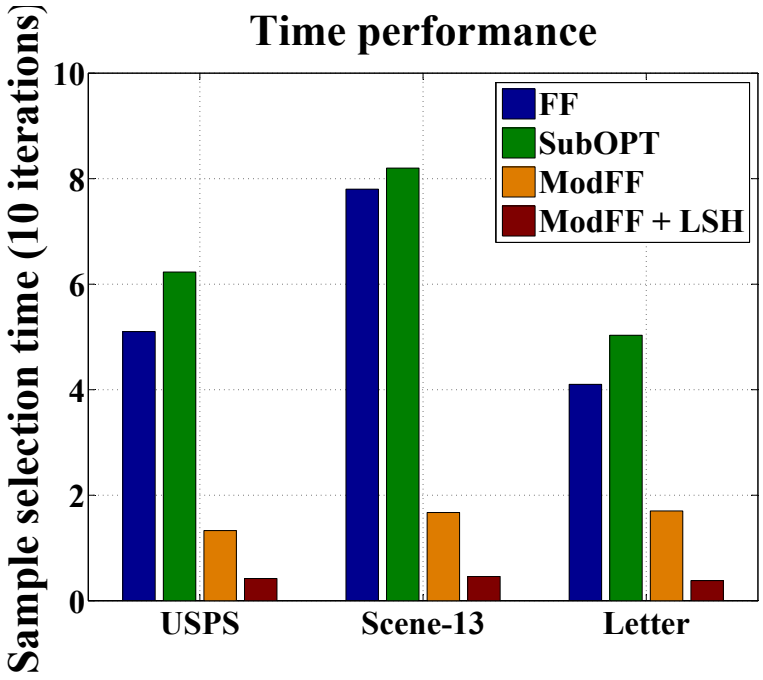


Figure 33: Time required for selection of samples from the active pool for different datasets. Note that even though farthest first (FF) and submodular optimization (SubOPT) are greedy algorithms, the quadratic scaling with respect to the batch size makes them slow. Modified farthest first is much faster due to the sub-sampling of informative samples that essentially reduces pool size. Using locality sensitive hashing results in a further order of magnitude speedup.

For USPS and Letter, we used 5000 samples in the unlabeled pool, and 1000 samples for testing. For Scene-13, 1000 samples were used in the unlabeled pool, along with another 1000 for testing. We experimented with k-NN for varying k between 1 and 10. The figures show results obtained using the 1 nearest neighbor classifier, however, other results are very similar.

Figure 32 shows classification accuracy values as a function of batch size, for two different datasets. We compare the random selection, greedy farthest first algorithm (FF), greedy submodular optimization algorithm (SubOPT), and the modified farthest first incorporating model information (ModFF). Note that the modified algorithm performs significantly better than the two greedy algorithms, which themselves beat random selection. This shows that active learning can be used to chose informative samples to achieve higher classification accuracy with the same training effort.

Finally, we show time results of the 4 different methods (3 methods shown above) along with the addition of the LSH approximation in Figure 33. As mentioned before, incorporating model information has the desirable side-effect of reducing the search space for the greedy algorithm, thus resulting in improved performance. The LSH method further speeds this up by at least 1 order of magnitude. Also note that this approximation is even more useful for larger datasets due to the asymptotic improvements in running time.

8.10 Iterative versus batch-mode selection

So far, we have discussed ways to actively select batches of informative and diverse samples, and studied how they outperform random batch selection. Here we reiterate that iterative active learning and batch-mode learning are inherently applicable to different problem settings.

Iterative active learning belongs to the realm of sequential design, while batch-mode selection is similar to *a-priori* selection. In principle, iterative learning receives information after each sample is chosen (i.e., the true label is revealed). However, batch selection must be done without any such intermediate information. Purely from this standpoint, we expect batch-mode selection to be weaker than iterative learning,

since it does not receive any information during the selection process.

Indeed, experiments confirm our belief – iterative learning with even random selection is better (and at worst competitive) with comprehensive batch-mode selection strategies for *improving classification accuracy* specifically. As such, whenever possible to perform iterative retraining and labeling, it is beneficial to work in the iterative setting which is more suitable for active learning in the classification domain. On the other hand, batch-mode learning is more suited for sensor placement and *a-priori* experiment design objectives. For theoretical aspects of the differences between sequential and *a-priori* design, see [Krause, 2008].

9 Application: Active Learning of Compact Hash Functions

Consider applications like image search and content-based image retrieval. Due to the typically large scale of these applications, one popular way to perform search and retrieval is to obtain image hashes that respect perceptual image similarity. Images can then be stored compactly by their hashes which often require orders of magnitude lesser space. Even more importantly, approaches such as locality-sensitive hashing (LSH) [Indyk and Motwani, 1998; Datar et al., 2004] allow finding nearest neighbors (for applications like image search) in time scaling much better than the naive approach that requires a linear scan over all the data. Typically, approximate hashing algorithms provide logarithmic or constant expected time retrieval of nearest neighbors.

Note that efficient data structures such as kd-trees [Bentley, 1975] and spill trees [Liu et al., 2004] have also been proposed for the purpose of searching exact and approximate nearest neighbors in high dimensional problems. However, our focus here is on hashing-type algorithms for near-neighbor search, hence we do not discuss these data structures further.

There has been a large amount of work on near-neighbor hashing in Hamming space [Charikar, 2002], Euclidean space [Datar et al., 2004], and many other metric spaces of interest. For images however, feature descriptors often do not lie in any of these spaces, and require complex distance function computations for estimating image similarity. Further, the similarity function is heavily application dependent, so the same feature descriptor might not be useful in all contexts. As such, researchers have explored many approaches such as metric embeddings to capture image similarity [Raginsky and Lazebnik, 2009], kernelized locality sensitive hashing [Kulis and Grauman, 2009], etc.

Although LSH-type approaches provide substantial speedups for near-neighbor

search compared to naive linear scans, they suffer from the drawback that many bits are required for sufficient discrimination of non-similar data samples. Increasing the number of bits in the hash code exponentially reduces the probability of true matches having the same code – in order to overcome this problem, many sets of hash functions are used in applications. This leads to undesirably large memory and query time requirements.

To address the issues, there has been recent interest in learning compact binary hash codes that incorporate perceptual similarity and are good are class discrimination, while at the same time have fewer bits than standard LSH-based codes [Salakhutdinov and Hinton, 2007b,a; Weiss et al., 2008; Raginsky and Lazebnik, 2009; Wang et al., 2010a; Lin et al., 2010; Wang et al., 2010b]. The main idea is to use data dependent projections instead of random projections. When labeled data is available, one can only focus on projection directions that achieve discrimination *according to the sample data*. Data dependent projections result in more compact codes since all projection vectors are targeted towards achieving discrimination, and none of them are "wasted" without providing data-specific information.

To further improve code compactness, in this chapter, we investigate how active learning can help learn discriminative functions with very few bits. We apply uncertainty-based active selection algorithms that learn hash functions using chosen labeled samples that are hardest to classify. Consequently, the learned functions provide good class discrimination with projection directions specifically optimized for them. This process results in a compact binary code that performs classification and retrieval with accuracy comparable with randomized codes (such as in standard LSH) as well as passively learned data-dependent codes of much larger size.

The two main aspects of interest are the approach for learning the hash functions, and the complementary one of actively selecting samples so as to quickly learn informative codes.

9.1 Learning hash functions

In this work, we restrict our attention to hash functions of the linear projection form: $f(x) = w^T \cdot x$. The actual hash bit is obtained by thresholding such that $h(x) = 0$

iff $f(x) < 0$ and $h(x) = 1$ otherwise. This formulation makes learning the hash codes easy, and is very similar to random projections as in LSH. Given a bunch of training samples, and large amounts of unlabeled data, we follow the semi-supervised learning scheme given by Wang et al. [2010b]. The following gives a brief overview of their approach.

Suppose we have a set of points $\mathbf{X} = [x_i], i = 1, \ldots, n$, with each x_i belonging to d-dimensional Euclidean space. The training data forms pairwise relationships between data points. If two points belong to the same category, they are *similar* and are denoted as belonging to conceptual class $\mathcal{M}$, containing similar pairs. Points not in the same class are *dissimilar*, and belong to the conceptual class $\mathcal{C}$. Note that $\mathcal{C}$ and $\mathcal{M}$ are classes of sample-pairs, and not individual samples themselves.

Given training data, we can identify these pairs from the class labels, and place them in appropriate sets. For notational convenience, the matrix $\mathbf{X}_L \in \mathbb{R}^{L \text{x} L}$ denotes the training data matrix.

Our goal is to learn a family of binary hash functions $\mathbf{H} = \{h_i, \ldots, h_k\}$, which can be concatenated for efficient retrieval. While learning the functions, the goal is to minimize the errors on the training data, otherwise known as empirical risk minimization. The empirical risk objective is defined as the total number of incorrectly classified pairs of samples, by each hash function (bit):

$$J(\mathbf{H}) = \sum_k \left\{ \sum_{(x_i, x_j) \in \mathcal{M}} h_k(x_i) h_k(x_j) - \sum_{(x_i, x_j) \in \mathcal{C}} h_k(x_i) h_k(x_j) \right\}. \tag{40}$$

To avoid problems with non-differentiability due to the thresholding, the above objective function is relaxed by replacing the *sign* function, with the signed magnitude of the projection:

$$J(\mathbf{W}) = \sum_k \left\{ \sum_{(x_i, x_j) \in \mathcal{M}} w_k^T x_i x_j^T w_k - \sum_{(x_i, x_j) \in \mathcal{C}} w_k^T x_i x_j^T w_k \right\}. \tag{41}$$

For notational simplicity, denote by $\mathbf{S} \in \mathbb{R}^{L \text{x} L}$ the matrix incorporating label information in the pairwise format, so that $S_{ij} = 1$ iff the pair $(x_i, x_j) \in \mathcal{M}$, and

$S_{ij} = -1$ iff the pair $(x_i, x_j) \in \mathcal{C}$. If training data on the pair is unavailable, $S_{ij} = 0$. Assuming this notation, the above can be rewritten as

$$J(\mathbf{W}) = \frac{1}{2}\sum_k w_k^T \mathbf{X}_l \mathbf{S} \mathbf{X}_l^T w_k = \frac{1}{2}\text{tr}\{\mathbf{W}^T \mathbf{X}_L \mathbf{S} \mathbf{X}_L^T \mathbf{W}\}. \tag{42}$$

However, given a few sample pairs belonging to similar and dissimilar classes, minimizing empirical risk can lead to severe overfitting [Wang et al., 2010b]. To overcome this problem, unlabeled data is used for regularization.

One desirable property of the learned hash functions (apart from achieving similar codes for similar data points) is to have maximum entropy. Intuitively, this implies that the hash function *spreads* the data points well across all conceptual buckets, and thus provides a meaningful partition of the data. Without regularization, grossly imbalanced partitions of the data are possible, similar to clustering or segmentation without regularization.

Following the maximum entropy principle, it is desirable to have balanced partitions of the data, so that $\sum_{i=1}^{n} h_k(x_i) = 0$. This is intractable to ensure in the exact case where the hash functions are thresholded [Weiss et al., 2008]. Instead, Wang et al. [2010b] show that maximizing the variance of a bit is equivalent to achieving a maximum entropy partition, and thus ensures a relatively balanced partition. Consequently, bit variance can be used to regularize the above objective function for empirical error. The objective function reduces to:

$$J(\mathbf{W}) = \frac{1}{2}\text{tr}\left\{\mathbf{W}^T \left[\mathbf{X}_L \mathbf{S} \mathbf{X}_L^T + \mu \mathbf{X}\mathbf{X}^T\right] \mathbf{W}\right\}, \tag{43}$$

where μ controls the amount of regularization. Note that the above objective function uses labeled as well as unlabeled data for regularization, and falls in the realm of semi-supervised hash code learning.

Further orthogonality constraints are often imposed on $\mathbf{W}$ such that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. Orthogonality ensures decorrelation of sequentially learned bits in the code, and is similar to PCA. The solution to the constrained problem is directly given by the

eigenvectors corresponding to the largest eigenvalues of the matrix

$$\mathbf{M} = \mathbf{X}_L \mathbf{S} \mathbf{X}_L^T + \mu \mathbf{X}\mathbf{X}^T. \tag{44}$$

Obtaining the solution involves an eigenvalue decomposition, and does not require computationally expensive iterative algorithms.

We use the formulation described above for learning hash codes given labeled and unlabeled data. The following describes how active selection is incorporated in order to chose samples so that compact codes can be learned.

9.2 Active selection

Restricting to the passive case, the above formulation still improves upon standard LSH due to its dependence on the (labeled) data. We wish to further compress the code length to achieve fast retrieval while requiring very little storage and hash code comparison time.

The idea of active selection for reducing sample complexity can directly be applied in this setting. The goal is to minimize the code length (i.e., learn a shorter but *discriminative* code), by intelligently selecting samples on which labels are requested. We focus on the principle of uncertainty sampling, noting that other measures are also applicable. To the best of our knowledge, this is the first application of active selection to the task of learning compact binary codes for classification and retrieval.

9.3 Uncertainty sampling

The goal is to chose training samples for hash learning in the semi-supervised framework described above, so that i) powerful discriminative codes are learned using a smaller initial training set (compared to randomly selecting samples for learning the code), ii) given the same training resources, one can use fewer bits to get the same classification / retrieval performance. Typically, both of these objectives can be achieved simultaneously, i.e., the former situation occurs when we can get away with using a smaller training set size to achieve a desirable accuracy rate, while the latter occurs when given a fixed classification rate, we can get away with a smaller code size.

We use the multi-class margin based BvSB uncertainty measure here again, i.e., we chose training samples that are the most uncertain for a classifier such as SVM given the previously training data, and append them along with their labels. This process continues until the required training resources are exhausted. As a comparison, we also use randomly selected training samples. Hash codes are learned using the described learning method above. We then experiment using this code for classification, i.e., the entire classification experiment is run by transforming each sample data point into the Hamming space by the learned functions. Intuitively, the code transforms the data into a much lower dimensional space in which classification can be performed faster, and more data can be stored. Furthermore, on similar lines nearest neighbor retrieval can also use the generated codes for a 'similar image search' type application.

Figure 34 shows the result using a starting set of 100 samples, and increasing the training set size to 200, in intervals of 10. We perform 20 runs with randomly chosen initial set, and the results show the average value.

It is important to note that in the experiment, the training data size *does not* refer to the data size used for training the final classifier. The size refers to the number for labeled samples used for *learning the hash code itself*. The classification is later performed on a fixed training set size of 50. We can thus conclude that choosing samples active lets the system *learn a better code* that captures the problem structure better, thus aiding in future classification. 20 bit code length was used for this experiment, although similar results were obtained for other code sizes as well (we experimented from 10 bits to 100 bit long codes).

In order to evaluate how the learning process affects hash code sizes, we do the following test. We experiment with both random and active learning of codes, and note the number of bits in the code when both methods achieve the same (very close) classification accuracy values. Table 5 shows the results, with accuracy values, and number of bits required using both methods on the USPS dataset. The experiment shows that active learning can aid in better allocation of storage and computation resources in realistic large-scale applications. Note that the passive process described here is from [Wang et al., 2010b], which itself outperforms random projection-based LSH for retrieval, since the former is data dependent, while the latter is not. **Our**

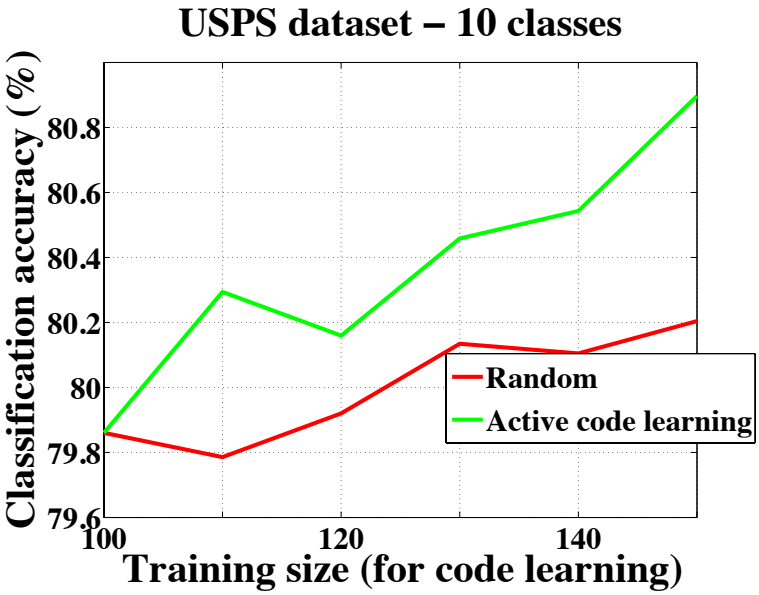


Figure 34: Active learning of hash codes for classification. The code is learned using the training set sizes on the X-axis. Final classification is performed by SVM on a fixed training set size (50 samples). The classification is performed in the domain of the learned code, i.e., all samples from the data are transformed by the hash functions learned into their corresponding Hamming space representations. SVM works directly in this Hamming space. For this experiment, we used a hash code length of only 20 bits.

	72%	75%	78%	82%
Random	6	18	38	75
Active	6	15	28	55

Table 5: Number of bits required by passive hash learning and active learning to achieve a given classification accuracy. Note that active learning of hash functions results in much more compact codes, thus aiding in improving speed and minimizing storage, while maintaining discriminative performance.

proposed algorithm thus improves upon an extremely competitive baseline that learns powerful data dependent projection directions.

In summary, the experiments illustrate that active selection can aid in learning hash codes that are discriminative, and are significantly more compact that standard random projection-based hashing algorithms as well as data dependent, learned hash functions. These results show that active selection for learning hash functions that are explicitly optimized for compactness are promising ways for fast nearest neighbor search and classification in image data, even when semantic similarities between two images are generally hard to capture.

10 Application: Incremental Learning in Evolving Scene Conditions

Most learning methods for detecting or classifying objects perform well when training and testing is done in similar conditions, such as on the same scene. However, conditions often change since training and deployment can be in different locations with widely varying illumination, camera position, apparent object sizes, pose of the subject/object. The generalization ability of trained classifiers is compromised in the presence of such changes. For example, Figure 35 shows the output of a human detector on a test frame from the CAVIAR dataset[6], when trained using Histogram of Oriented Gradients (HOG) features [Dalal and Triggs, 2005] on a subset of the INRIA pedestrian dataset.

For each frame of test video, we employ a sliding window of 75 pixels by 50 pixels wide, with horizontal and vertical overlap of 50 and 30 pixels respectively. HOG features are extracted for each window, and the obtained vector is passed through the trained Support Vector Machine classifier. Red boxes indicate a positive classifier output, i.e., the particular bounding box contains a human according to the trained classifier. The figure shows an extremely large number of false positive detections, primarily due to misleading texture in the upper part of the frame.

From the previous example, it is clear that the learned human models *do not generalize well*, and heavily rely on the specifics of the training data. The background texture is never seen in training, and is consequently classified as a human in the new frame. On the other hand, we can also see that the human is detected correctly in the frame. The model therefore correctly captures some aspects of the detection problem, specifically, the appearance of the human.

Motivated by the partial correctness of the learned model, our objective is to *adapt*

[6]http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

(a) (b)

Figure 35: (a) Original frame. (b) Output of a human detector trained on the INRIA dataset.

it to the new scene efficiently and quickly, i.e., with little or no user input. The goal is to retain the informative aspects of the previous training data, while also gathering more information about the new classification task, thereby constructing a scene-specific classifier from a generic one. In this chapter, we focus on the application of human detection, which has been an area of significant recent research [Orrite-Urunuela et al., 2004; Dalal and Triggs, 2005; Tuzel et al., 2007; Zhu et al., 2006; Haga et al., 2004; Hussein et al., 2009]. However, note that the approach can be applied to other detection tasks as well.

Broadly, our method works via performing incremental updates by *actively selecting* new instances for training and removing old uninformative instances. The removal of training examples allows us to maintain fixed training sizes, so training is efficient, and can work on a fixed memory budget. This is particularly important for installing embedded software in the cameras themselves which is gaining a lot of interest due to the potential for easy deployment.

Consider the following setting. We have access to a large set of training examples from a standard dataset, such as INRIA pedestrian data (generic data). The objective is to deploy a classifier (human detector) on a new scene wherein we can access frames from the video sequence captured by the camera. We propose two modes of system operation. The first mode is that of semi-supervised adaptation, with user in the

learning loop. The system adapts to the new scene based on a few queries made to the user (such as showing an image window and querying whether it consists of a human or not). In the second autonomous mode, the system uses generic data along with the first few frames from the new video (which does not contain any motion) to learn a scene-specific classifier. The first mode is for more challenging environments where human appearance may differ significantly or where empty frames are not available for autonomous adaptation.

10.1 Adaptation with user in the loop

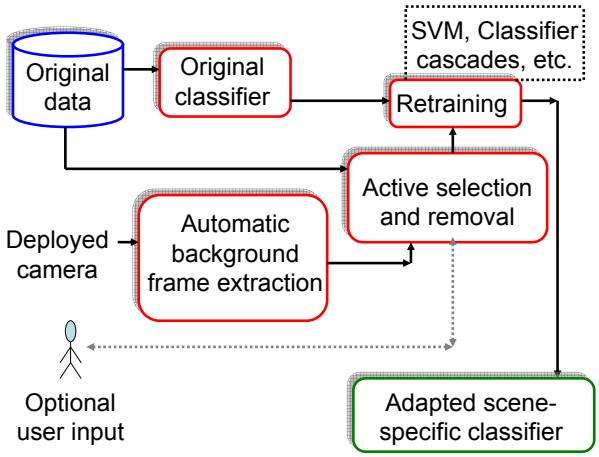


Figure 36: Block schematic of the proposed system.

10.2 Active learning

Here, we employ active learning for choosing samples to be annotated by the user, so that adaptation to the new environment can be done very fast, with the help of only a few queries. We use SVM classifier, and the uncertainty measure is the one proposed in Chapter 4, i.e, the BvSB measure, which, in the binary setting, reduces to margin from the classification boundary. In our first approach, uncertainty

sampling is used along with SVM for adding new samples to the training set. In order to maintain fixed training set sizes (for embedded applications having memory and speed constraints), we also remove "irrelevant" or "uninformative" samples from the training set as described later in the chapter.

The following section describes Passive-Aggressive (PA) algorithms that are traditionally used for online learning. We observe that active selection with SVM performs much better for our application that active selection along with PA classifiers, however we include a review for completeness.

10.3 Online passive-aggressive algorithms

In this section, we provide a brief review of passive aggressive (PA) algorithms [Cramer et al., 2003] used for fast incremental learning. PA algorithms are online incremental algorithms that sequentially modify the classifier (analogously, regresser) after receiving the label on a sample.

The setting is as follows. The classifier is defined by a linear projection based on a weight vector of the form $f(x) = sign(w^T \cdot x)$. In the realizable case[7], the optimal vector w^* for a set of samples is the one which incurs zero loss $\mathcal{L}(w^*; x, y)$ for all sample pairs (x, y). For correct prediction above a margin ϵ, no loss is incurred, whereas otherwise, the loss incurred is $\epsilon - y(w \cdot x)$.

In the online setting, the weight vector is denoted as w_t parameterized by iteration number or time t. Given a new training sample x_{t+1}, the classifier predicts its label using the current w_t. After the prediction, the true label is revealed, and the classifier suffers an instantaneous loss $\mathcal{L}(w_t; x_{t+1}, y_{t+1})$. For correct prediction above a margin ϵ, no loss is incurred, whereas otherwise, the loss incurred is $\epsilon - y_{t+1}(w_t \cdot x_{t+1})$. The update rule defined by Cramer et al. [2003] is

$$w_{t+1} = \underset{w}{\operatorname{argmin}} \|w - w_t\|, \text{ s.t. } \mathcal{L}(w; x_{t+1}, y_{t+1}) = 0. \tag{45}$$

In other words, the new weight vector is obtained by projecting the current vector into the space of all vectors that incur no loss. In the classification setting, the space

[7]Realizable case is one that admits the existence of a vector that incurs zero loss.

of all vectors that incur no loss is a half-space $C = \{w : -y_{t+1} w_t \cdot x_{t+1} \leq \epsilon\}$.

The above optimization problem has a closed form solution given by [Cramer et al., 2003]

$$w_{t+1} = w_t + \tau_t v_t, \tag{46}$$

where $\tau_t = \mathcal{L}(w; x_{t+1}, y_{t+1})/\|v_t\|^2$, and $v_t = y_{t+1} x_{t+1}$.

The classifier remains *passive* when classification on the new sample is correct and no loss is incurred, while it *aggressively* tries to eliminate the loss otherwise. For the unrealizable case, the loss function is modified to consider the loss *relative* to the minimum achievable loss. This results in only a modification to the update rule above so that

$$\tau_t = \frac{\mathcal{L}(w; x_{t+1}, y_{t+1})}{\|v_t\|^2 + \gamma},$$

where γ is a relaxation parameter. In the realizable case, Cramer et al. [2003] prove a mistake bound derived from bounding the total loss after a set of iterations. Analogously for the non-realizable case, the total loss relative to the minimum achievable loss is bounded.

The PA algorithm is well-suited to the task of online classification in conditions where data drifts. In our application, we would like to have a system that continuously tracks changes in the environment to adaptively adjust the classification rule to provide correct classification. Thus, we also use the PA algorithm along with active selection to develop an incremental learner. One advantage is that this works well in the resource-constrained setting since it does not involve keeping all training samples, but only the current vector instead. Performing the update is fast due to the closes form computation, and errors are reversed quickly.

However, there are also a couple of limitations to this approach. The first limitation is that it cannot be kernelized easily – many image features are not vector-space representations, and similarity computations often involve kernel function evaluations. Since the PA algorithm explicitly keeps the projection vector, it cannot be used with similarity kernels. Second, since the algorithm involves only keeping the

current classifier, recurrent themes might not be captured easily. For instance, classifiers need to adapt for night and day conditions, which occur repeatedly. In such a scenario, previous training samples might turn out to be valuable in the future. SVM incremental learners that we propose overcome both of these limitations effectively, while compromising on retraining time. However, since updates are rarely performed, the tradeoff is extremely suitable for our application.

10.4 Incremental learning and forgetting

In this section, we employ active learning and forgetting for incremental learning. The main idea is that given a set of *generic* training images, new informative images *from the location of deployment* can be queried to the user for adding to the training set, while old uninformative images can be removed. The selection and deletion (forgetting) processes both work through active selection. For deletion, the active selection measure is inverted – i.e., examples which are least informative are selected. This simple mechanism results in extremely fast classifier updates, and there is negligible difference in accuracy incurred due to lost training samples, primarily since the most uninformative ones are removed.

Figure 36 shows our learning setup. Given a new scene for deployment along with generic training data, the method queries the user and adds a few training images from the new frame. This little training data allows the classifier to quickly adapt to the new scene. At the same time, old uninformative data is removed from the training set, thus limiting the total memory and training time. As the examples to be removed are selected actively, they are relatively uninformative and the removal does not significantly hurt accuracy. This process is performed iteratively, and it results in a new classifier that is scene-specific, achieved by adapting the generic training data with little human input. In general, at a new deployment location, the first few frames of video can be used for performing the update, and the resulting classifier can then be deployed on the location.

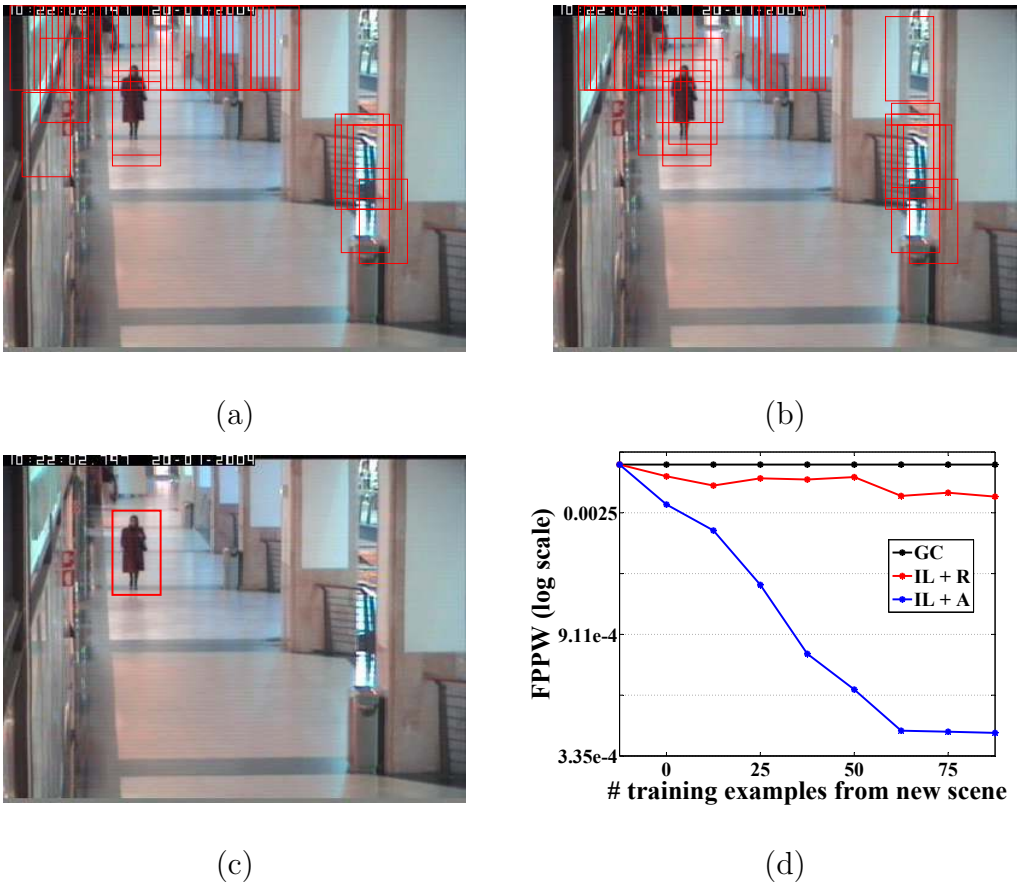


Figure 37: Sample results with 75 training examples from CAVIAR, (a) Generic classifier, (b) Incremental learning with random selection, (c) Incremental learning with active selection. (d) FPPW values of different methods with varying number of training examples on VID.

10.5 Results with semi-supervised adaptation

The experiments are performed on two video sequence, one obtained from a local parking lot (referred to as VID) and another from the CAVIAR dataset. The extracted image frames have a lot of confusing reflections and texture.

Since the mistakes in predictions are false positives, the primary evaluation measure used is False Positives per Window (FPPW) for each frame. All the classifiers detect the human in the frame correctly.

In this section, we compare our method of incremental active learning (IL+A) with two baselines: i) using the generic classifier on VID (called GC), and ii) incremental learning, but with Random selection of examples to add and remove, instead of active selection (IL+R). Figure 37(d) shows the achieved FPPW rate over multiple frames on VID, alongside the number of training examples used from VID. Figures 37(a)–(c) shows sample detection results on one frame of CAVIAR. The improvement of incremental active learning over the generic classifier demonstrates the importance of scene-specific training, whereas the improvement over random selection demonstrates the importance of active selection.

Note that our proposed method is not intended to replace other detection techniques, but rather *complement* them by adding incremental active learning. As such the proposed approach can be used with other existing techniques that perform well in particular domains, such as classifier cascades which have been demonstrated to give good performance in human detection applications [Hussein et al., 2009].

The above method of semi-supervised adaptation can be applied to many incremental learning tasks, even when training and test conditions differ substantially and no other information is available. In many real human detection applications, more information is available. For example, at the deployment location, we might have access to a few frames of video without any human in the scene. Alternatively, motion sensors are often available in surveillance environments – these motion sensors can be used as a primary sensor to indicate the presence of a frame without a human. In these scenarios, we can adapt the generic classifier to the new scene completely autonomously as follows.

10.6 Autonomous adaptation

In the example of Figure 35, there are a large number of false positives we aim to eradicate, while keeping the correct detection as is. If we have access to the video frames when there is no human in the scene, we can use image windows from that frame to gather more *negative* training images.

10.6.1 Which negative examples to select?

The number of sliding windows per frame can be very large, because of the small window size and substantial overlap. As such, it is impractical to use all of the windows as negative training instances, from both perspectives of training set size, and retraining time. In this section, we discuss our method of example selection and removal.

The generic classifier is deployed on the empty frame, and all the windows on which it gives a *positive* response are selected for training. As the frame is known to be empty, the positive detections are essentially **misclassifications** by the generic classifier. Therefore, adding them to the training data is likely to change the classifier, and reduce the number of false positive detections.

10.7 Maintaining training set sizes

On the other hand, adding new training instances increases the size of the training set. This is undesirable in memory-constrained situations and where processing rate is critical. Therefore, we also propose to *remove* an equal number of old negative examples from the generic data. This is accomplished by using the method of the previous section, i.e., removing examples that are farthest away from the boundary.

10.8 Results with autonomous adaptation

Figure 38 shows the results of using initial background frames to extract false negatives along with the generic training data. As the number of background frames used increases, the number of false positives goes down. The method is thus a viable candidate to adapt a classifier to a new location *without the need for any human*

(a) (b)

Figure 38: (a),(b) show results with using 1 and 2 background frames respectively for autonomous updates.

supervision. Furthermore, in many cases scene conditions such as illumination levels change over time. One can use autonomous updates using empty frames to adapt the classifier when such changes occur, so that detection quality is consistently maintained.

To summarize, we proposed two approaches, one completely autonomous, and one with little user supervision to adapt generic training data to provide scene-specific detectors. The discussed methods address the important issue of quick deployment in various locations, without involving expensive operations of data collection at the location. Using incremental learning, the classifiers can combine the advantages of available generic data as well as scene-specific data, and the small memory footprint is particularly suitable for embedded applications such as when detection software is on a camera.

11 Conclusions

In this thesis, we have explored ways to perform image classification and search via ideas in active learning. The main goal was to reduce the training and supervision required in image classification applications, which have traditionally suffered from needing a large amount of training data.

Specifically, we focus on the uncertainty sampling method of active learning, with some algorithms based on other measures such as coverage and decision theoretic risk minimization. The proposed algorithms are directed towards large multi-class problems with hundreds of categories, and huge data sizes. We observe that useful notions of uncertainty for multi-class problems are substantially different from binary uncertainty measures. Due to the presence of multiple classifiers, it is essential for multi-class active learning to select samples from the weakest classifiers.

One of the biggest bottlenecks in training large multi-class systems is giving precise labels to queries. Such a process is interactively inefficient. In order to overcome the problem, we demonstrate a binary feedback modality for multi-class classification. Experiments show that such a feedback modality substantially reduces training time requirements, and allows distributed annotation on a very large scale. Scalability is another primary concern in active learning, and a we have strived to make approaches computationally efficient and scalable throughout the thesis.

Finally, we note that active learning is applicable in a wide variety of scenarios, including sensor placement, efficient incremental learning, and learning of compact hash codes for image retrieval.

11.1 Contributions of the thesis

- We propose a simple multi-class active selection measure that selects uncertain samples in the presence of hundreds of classes. The measure is easy to compute, and automatically selects samples for the weakest classifiers.

- A binary feedback method is demonstrated for learning multi-class classifiers, that substantially reduces training time and effort, and allows large-scale distributed annotation across many users.
- Active learning formulations considered in the literature have focused on either query synthesis or selective sampling – we combine those approaches so as to retain advantages of both. In other words, the method performs efficient selection similar to query synthesis, and still choses samples from the unlabeled pool to allow meaningful queries.
- Locality-sensitive hashing approaches are used for efficient near neighbor search, which is combined with the above to find unlabeled samples close to the synthesized query. LSH provides sublinear time scaling with respect to the pool size, allowing fast active learning with over a million samples on a single computer.
- When iterative training is not practical, batch-mode active learning is essential, i.e., users are queried for labels of a large batch of samples simultaneously. We propose greedy algorithms for efficient batch selection which outperform random selection in classification accuracy.
- Finally, we demonstrate two important applications of active learning: i) active learning for generating extremely compact codes for efficient image retrieval. The main idea is to chose samples actively so as to learn codes that are more compact that passively learned ones, providing advantages for fast similarity search. ii) incremental learning for training human detectors that adapt to changing environmental conditions such as lighting, etc. Active selection helps chose samples that the current classifier misclassifies, and then updates itself through human supervision or autonomously with the help of other sensors.

11.2 Future work directions

The results in the thesis demonstrate the promise of active learning and open up many new directions for future work.

- Exploiting class correlations in multi-class and multi-label problems. Currently, we focus on each class independently, and perform the classification separately. However, in reality, correlations between the different classes often exist since some objects tend to co-occur in images. Also, hierarchical categorization of object categories is possible. While we have not explored these aspects here, there is potential to improve classification performance by considering the correlations and hierarchies, thus going towards semantic analysis of image content.

- There is a lot of recent interest in parallel algorithms, primarily due to the success of distributed computing systems such as MapReduce [Dean and Ghemawat, 2004]. Although some of the algorithms proposed in the thesis can be distributed and implemented parallely, there is scope for further research on explicitly exploiting parallelism. We believe parallel computation can be particularly useful in the active learning domain due to the availability of crowdsourcing for achieving labeling. Consequently, parallel computation as well as parallel interaction modalities need to be explored.

- Throughout the thesis, we have focused on minimizing the misclassification rate of a trained classifier. It is conceivable that there are domains where other objectives are more relevant than the misclassification rate. In order to actively learn in such domains, other objective functions directly related to the application need to be considered. Further, explicit objective functions might not even be available in many cases involving decision making, wherein ways to learn the functions themselves need to be explored along with active learning for minimizing training.

- There are many real-world applications involving statistical learning and data analysis that can benefit from active learning. We have primarily focused the work in the realm of image classification, however, most of the proposed methods are much more generally applicable. Research in adapting the methods to specific applications can potentially have a high impact in improving performance, while allowing further automation.

In summary, since human time is the most precious element in large-scale learning systems, active learning has a lot of potential for simplifying learning tasks in a wide variety of domains. The experiments in this thesis show very promising results for using active learning, and we believe research in this direction has a lot of potential for developing large-scale autonomous systems that improve continually.

References

Amazon mechanical turk. http://www.mturk.com.

D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

D. Angluin. Queries revisited. In *Proceedings of the International Conference on Algorithmic Learning Theory*, 2001.

V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56:411–421, 2006.

A. Asuncion and D. J. Newman. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2007. Available at http://archive.ics.uci.edu/ml/datasets.html.

M.-F. Balcan, A. Blum, P. P. Choi, J. Lafferty, B. Pantano, M. R. Rwebangira, and X. Zhu. Person identification in webcam images: an application of semi-supervised learning. In *ICML Workshop on Learning with Partially Classified Training Data*, 2005.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *International Conference on Machine Learning*, 2006.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Conference on Computational Learning Theory*, 2007.

T. Bandos, D. Zhou, and G. Camps-Valls. Semi-supervised hyperspectral image classification with graphs. In *Proceedings of the IEEE Geoscience and Remote Sensing Symposium*, 2006.

S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, 2002.

M. Belkin, I. Matveeva, and P. Niyogi. Tikhonov regularization and semi-supervised learning on large graphs. In *International Conference on Acoustics, Speech, and Signal Processing*, 2004.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

A. Berg and J. Malik. Geometric blur for template matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *International Conference on Machine Learning*, 2009.

R. Bhatia. Positive definite matrices. *Princeton Series in Applied Mathematics*, 2007.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *International Conference on Machine Learning*, 2001.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory*, pages 92–100, 1998.

A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

K. Brinker. Incorporating diversity in active learning with support vector machines. In *International Conference on Machine Learning*, 2003.

C. Campbell, N. Cristianini, and A. J. Smola. Query learning with large margin classifiers. In *International Conference on Machine Learning*, 2000.

K. Chaloner and I. Verdinelli. Bayesian experimental design: a review. *Statistical Science*, 10:237–304, 1995.

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

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the ACM Symposium on Theory of Computing*, 2002.

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

K. Cramer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

K. Cramer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. In *Advances in Neural Information Processing Systems*, 2003.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*. MIT Press, 2005.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*. MIT Press, 2006.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *International Conference on Machine Learning*, 2008.

S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Conference on Computational Learning Theory*, 2005.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*. MIT Press, 2008.

M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on computational geometry*, 2004.

J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Sixth Symposium on Operating System Design and Implementation*, 2004.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A Min-max cut algorithm for graph partitioning and data clustering. In *IEEE International Conference on Data Mining*, 2001.

P. Dollár, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

S. Ertekin, J. Huang, L. Bottou, and L. Giles. Learning on the border: active learning in imbalanced data classification. In *ACM International Conference on Information and Knowledge Management*, 2007.

Facebook, 2010. http://www.facebook.com/press/info.php?statistics.

V. Federov. *Theory of optimal experiments*. Academic Press, 1972.

D. Fehr, A. Cherian, V. Morellas, and N. Papanikolopoulos. Compact covariance descriptors in 3D point clouds for object recognition. Under review in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

L. Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

A. Ferencz, E. G. Learned-Miller, and J. Malik. Learning to locate informative features for visual identification. *International Journal of Computer Vision*, 77:3–24, 2008.

R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *IEEE International Conference on Computer Vision*, 2005.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems*. MIT Press, 2007.

L. Gomez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe. Semisupervised image classification with laplacian support vector machines. *IEEE Geoscience and Remote Sensing Letters*, 5:336–340, 2008.

K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision*, 2005.

K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *International Joint Conference on Artificial Intelligence*, 2007.

T. Haga, K. Sumi, and Y. Yagi. Human detection in outdoor scene using spatio-temporal motion analysis. In *IEEE International Conference on Pattern Recognition*, 2004.

S. Hanneke. *Theoretical foundations of active learning.* PhD thesis, Carnegie Mellon University, 2009.

S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semi-supervised SVM batch mode active learning for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *International Conference on Machine Learning*, 2006.

A. Holub, M. Welling, and P. Perona. Exploiting unlabelled data for hybrid object classification. In *NIPS 2005 Workshop on Inter-Class Transfer*, 2005.

A. Holub, P. Perona, and M. Burl. Entropy-based active learning for object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshop on Online Learning for Classification*, 2008.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.

M. Hussein, F. Porikli, and L. Davis. A comprehensive evaluation framework and a comparative study for human detectors. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):417–427, 2009.

P. Indyk and R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. In *Proceedings of the Symposium on Theory of Computing*, 1998.

P. Jain and A. Kapoor. Active learning for large multi-class problems. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

P. Jain, S. Vijayanarasimhan, and K. Grauman. Hashing hyperplane queries to near points with applications to large-scale active learning. In *Advances in Neural Information Processing Systems*, 2010.

V. Jain, A. Ferencz, and E. G. Learned-Miller. Discriminative training of hyper-feature models for object identification. In *Proceedings of the British Machine Vision Conference*, 2006.

F. Jing, M. Li, H.-J. Zhang, and B. Zhang. Entropy-based active learning with support vector machines for content-based image retrieval. In *ICME '04: IEEE International Conference on Multimedia and Expo*, 2004.

A. J. Joshi and N. Papanikolopoulos. Learning to detect moving shadows in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2055–2063, 2008.

A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with Gaussian Processes for object categorization. In *IEEE International Conference on Computer Vision*, 2007a.

A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*, 2007b.

R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Bryne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan, and A. Claire. The automation of science. *Science*, 324:85–89, 2009.

A. Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, 2008.

A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Uncertainty in Artificial Intelligence*, 2005.

A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. Technical Report CMU-ML-08-100, Carnegie Mellon University, 2008.

A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Robust sensor placements at informative and cost-effective locations. *ACM Transactions on Sensor Networks*, 7 (4), 2011.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *IEEE International Conference on Computer Vision*, 2009.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

K. Lang and E. Baum. Query learning can work poorly when a human oracle is used. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 1992.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *IEEE International Conference on Computer Vision*, 2003.

H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68:267–276, 2007.

R.-S. Lin, D. Ross, and J. Yagnik. SPEC hashing: Similarity preserving algorithm for entropy-based coding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27:986–1005, 1956.

T. Liu, A. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. In *Advances in Neural Information Processing Systems*, 2004.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

P. Melville. *Creating diverse ensemble classifiers*. PhD thesis, The University of Texas at Austin, 2003.

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

T. Mitchell. *Machine Learning*. Boston: McGraw-Hill, 1997.

K. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Information Processing Systems*. MIT Press, 2003.

I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *International Conference on Machine Learning*, 2002.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, 2000.

A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

A. Oliva and A. Torralba. Building the Gist of a scene: the role of global features in recognition. *Progress in Brain Research*, 155, 2006.

C. Orrite-Urunuela, J. M. del Rincón, J. E. Herrero-Jaraba, and G. Rogez. 2D Silhouette and 3D skeletal models for human detection and tracking. In *IEEE International Conference on Pattern Recognition*, 2004.

N. Panda, K. Goh, and E. Chang. Active learning in very large image databases. *Journal of Multimedia Tools and Applications: Special Issue on Computer Vision Meets Databases*, 31, 2006.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66:41–66, 2006.

J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 2000.

G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang. Two-dimensional active learning for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*, 2009.

R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In *International Conference on Machine Learning*, 2007.

R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighborhood structure. In *International Conference on Artificial Intelligence and Statistics*, 2007a.

R. Salakhutdinov and G. Hinton. Semantic hashing. In *SIGIR Workshop on Information Retrieval and applications of Graphical Models*, 2007b.

R. Segal, T. Markowitz, and W. Arnold. Fast uncertainty sampling for labeling large e-mail corpora. In *Conference on Email and Anti-Spam*, 2006.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Conference on Computational Learning Theory*, pages 287–294, 1992.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

R. Sivalingam, V. Morellas, D. Boley, and N. Papanikolopoulos. Metric learning for semi-supervised clustering of region covariance descriptors. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2009.

J. Sivic, B. Russell, A. Efros, and A. Zisserman. Discovering object categories in image collections. In *IEEE International Conference on Computer Vision*, 2005.

E. Sudderth, A. Torralba, W. T. Freeman, and A. Wilsky. Describing visual scenes using transformed dirichlet processes. In *Advances in Neural Information Processing Systems*. MIT Press, 2006.

E. Sudderth, A. Torralba, W. T. Freeman, and A. Wilsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, pages 291–330, 2008.

S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, 2001.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*. MIT Press, 2006.

A. Torralba, R. Fergus, and W. T. Freeman. 80 Million tiny images: a large database for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1958–1970, 2008.

O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on Riemannian manifolds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1713–1727, 2008.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

S. Vijayanarasimhan and K. Grauman. Multi-level active prediction of useful image annotations for recognition. In *Advances in Neural Information Processing Systems*. MIT Press, 2008.

S. Vijayanarasimhan and K. Grauman. What's it going to cost you? : Predicting effort vs. informativeness for multi-label image annotations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: training object detectors with crawled data and crowds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *International Conference on Machine Learning*, 2010a.

J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010b.

M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000a.

M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European Conference on Computer Vision*, 2000b.

Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, 2008.

D. Williamson and D. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2010.

T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *IEEE International Conference on Computer Vision*, pages 516–523, 2003.

H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136, 2006.

W. Zhao, J. Long, E. Zhu, and Y. Liu. A scalable algorithm for graph-based active learning. *Frontiers in Algorithmics*, 2008.

Z.-H. Zhou, K.-J. Chen, and Y. Jiang. Exploiting unlabeled data in content-based image retrieval. In *European Conference on Machine Learning*, 2004.

Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, 2003.