CARNEGIE MELLON UNIVERSITY

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

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

TOWARD EFFICIENT COLLABORATIVE CLASSIFICATION FOR DISTRIBUTED VIDEO SURVEILLANCE

Christopher Paul Diehl

ACCEPTED BY THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

John B. Hampshire II, MAJOR PROFESSOR DATE

Pradeep K. Khosla, DEPARTMENT HEAD DATE

APPROVED BY THE COLLEGE COUNCIL

John L. Anderson, DEAN DATE

CARNEGIE MELLON UNIVERSITY

TOWARD EFFICIENT COLLABORATIVE CLASSIFICATION FOR DISTRIBUTED VIDEO SURVEILLANCE

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY
in
ELECTRICAL AND COMPUTER ENGINEERING

by

Christopher Paul Diehl

Pittsburgh, Pennsylvania
December 2000

To Mom, Dad and Lisa

Thesis Committee

John B. Hampshire II, advisor
Department of Electrical and Computer Engineering
Carnegie Mellon University

Pradeep K. Khosla
Department of Electrical and Computer Engineering
Carnegie Mellon University

Vijayakumar Bhagavatula
Department of Electrical and Computer Engineering
Carnegie Mellon University

Barak A. Pearlmutter
Department of Computer Science
University of New Mexico

Acknowledgements

These last four years have been very enriching on many levels, both personally and professionally. I am deeply grateful to a number of people who have offered their guidance and support along the way.

First, I would like to thank my friend and advisor Dr. John Hampshire for introducing me to the field of computational learning. Our conversations over the years have been very enlightening. It has been a real pleasure to have the opportunity to explore this landscape of ideas with him.

I would also like to thank my committee members Professor Pradeep Khosla, Professor Vijayakumar Bhagavatula, and Professor Barak Pearlmutter for taking the time to provide me with their invaluable perspective.

My heartfelt thanks also goes out to Lynn Philibin and Elaine Lawrence in the ECE Grad Office. They are the heart and soul of this department. I thank them for their unending support of the grad students in ECE.

Thanks to Dr. Ashitey Trebi-Ollennu, Dr. John Dolan and all the members of the CyberScout team. It has been a pleasure working with them to build this state-of-the-art system.

I want to thank my friend and colleague Mahesh Saptharishi for all of his support. Without his assistance, many of these experiments would not have been possible.

Thanks to my friends Dr. Bill Spears and Dr. Diana Gordon at the Naval Research Lab. They have been my advocates from afar. I am deeply grateful for their support.

And finally I would like to thank my Mom and Dad and my sister Lisa for all their love and support over the years. This thesis is dedicated to them.

Christopher P. Diehl
Pittsburgh, Pennsylvania
December 14, 2000

Abstract

In this thesis, we propose a general strategy for automated video surveillance that relies on collaboration between the surveillance system and the user. Such collaboration enables the user to help the system incrementally acquire the necessary context for truly robust surveillance. The success of this strategy is dependent on the ability of the system to identify novel instances of known or unknown classes that it does not understand. This, in turn, allows the user to focus only on the observations with the highest uncertainty that require interpretation.

Designing a real-time classification process that supports novelty detection is nontrivial. The real-time constraint dictates computational simplicity, whereas novelty detection requires a high dimensional feature space to aid in discriminating between the known and unknown classes. The majority of this work focuses on the problem of simultaneously satisfying these conflicting constraints. We consider these issues in the context of a relevant surveillance task and evaluate the performance of the resulting classification process in the CMU Cyberscout distributed video surveillance system.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Need for Automated Video Surveillance

The objective of surveillance is to monitor a given environment and report information about relevant activity. *Video surveillance* typically involves using electro-optical sensors to observe the environment. Video surveillance systems have predominantly been employed to monitor activity within and around office buildings, airports and other facilities. A basic video surveillance system consists of a collection of video cameras mounted in fixed positions or on pan-tilt devices. The video streams are transmitted to a central location, displayed on one or several video monitors and recorded. Security personnel observe the video to determine if there is ongoing activity that warrants a response. Given that such events may occur infrequently, detection of salient events requires focused observation by the user for extended periods of time. Placing the burden of interpretation on the user imposes severe limits on the number of sensors that can be effectively utilized and the amount of information derived from the system.

Commercially available video surveillance systems attempt to reduce the burden on the user by employing *video motion detectors* to detect changes in a given scene [25]. Video motion detectors can be programmed to signal alarms for a variety of reasonably complex situations. Yet the false alarm rate for most systems in typical environments is unacceptable. In addition, programming the system to detect an event of interest is nontrivial and requires a significant amount of user training [25]. Obviously this provides the user with little additional benefit.

Ideally, a video surveillance system should only require the user to specify the objectives of the surveillance mission and the context necessary to interpret the video in a simple, intuitive manner. When in operation, the system should provide the user with timely assessments of relevant activity that allow the user to affect the outcome of an ongoing event. For many scenarios within the civilian and military sectors, real-time interpretation is required for the information produced by the system to be valuable. Therefore the challenge is to provide robust real-time video surveillance systems that are easy to use and are composed of inexpensive, commercial off-the-shelf hardware for sensing and computation.

Given the capability to interpret activity in video streams in real-time, the utility of a video surveillance system increases dramatically and extends to a larger spectrum of missions. With such a system, a single user can observe the environment using a much larger collection of sensors. In addition, continuous, focused observation of activity for extended periods of time becomes possible. As such capabilities mature, the roles of video surveillance systems will encompass activities such as peace treaty verification, border monitoring, surveillance of facilities in denied areas, hazard detection in industrial facilities and automated home security.

1.2 Understanding Activity in Video

Before one can begin to formulate a strategy for automating the video interpretation, it is important to first consider the objectives of the surveillance mission. Generally when monitoring a given area, we would like to answer the following questions.

- *Is there any change in the scene?*
- *If so, what objects are moving about the environment?*
- *Where are they located?*
- *Where are they going?*
- *What are they doing?*

When watching a video stream, how does one answer these questions? Each one of us has a vast store of knowledge that we've acquired from past experience. Using this knowledge base, we attempt to explain the video in terms of concepts we understand. We may also have additional knowledge about the scene which helps us focus only on the relevant possibilities. In short, we are applying context to derive the most likely explanation of the video data.

Automating this process involves defining a series of interpretation processes that incrementally transform the pixel level description to a qualitative, semantic level description of the activity. At each step, context is applied to achieve the transition. Context is generally specified in the form of a parameterized representation and associated procedures for tasks such as parameter adaptation and model evaluation.

Given the challenge of achieving real-time performance, system designers must give careful thought to the types of representations employed. Typically the system designer specifies and fixes the representations and the associated interpretation processes prior to the deployment of the system so that the performance of the system can be optimized for a particular mission. This implies the system is restricted to interpreting activity in the environment in terms of the original context specified. When surveillance systems are deployed in uncertain or changing environments, the necessary context to interpret the environment cannot be completely defined beforehand. The system must have some mechanism to incrementally learn context from the user.

1.3 Collaborating with the User: The Interpretation Cycle

When designing a classifier, the first step is to label a large set of examples so that the classifier parameters can be estimated from the data during training. Such a process can be very time intensive and may not be practically feasible especially once the system is operational. What is needed is a procedure that allows the user to efficiently review observations made by the system, detect novel, informative events and update the classifier using examples labeled by the user.

When the system is operational, we envision the following cycle of collaboration between the user and the system to achieve this objective. We have termed this process the *interpretation cycle* which is illustrated in figure 1.1. At the beginning of the surveillance mission, the user interactively labels a limited number of examples of events to be classified by the surveillance system. Once the labeled examples are specified, the system designs a classifier and begins to observe and interpret the environment in real-time. Since the system is assumed to have incomplete knowledge of the environment, it evaluates its confidence in classifications in order to identify novel instances of known classes or unknown classes. After

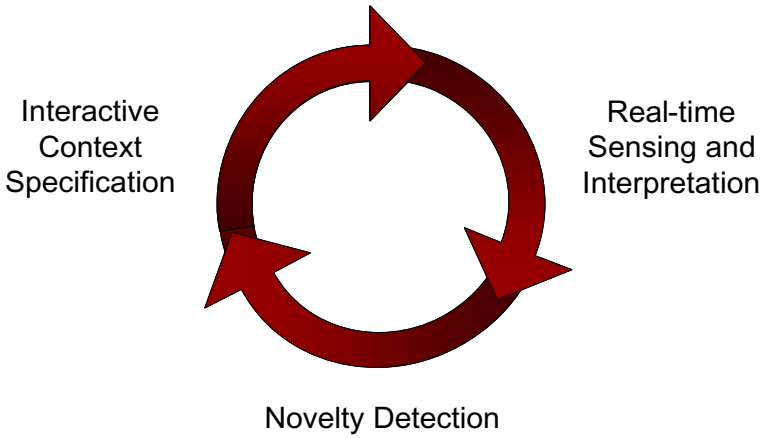


Figure 1.1: The interpretation cycle

a certain amount of time, the user is asked to review and label some of the observations that have the highest uncertainty. If a set of observations do not correspond to any of the specified classes, the user may establish a new class. Using the additional labeled data, the system reconfigures the classifier and the cycle continues.

1.4 Implementation Challenges

In order to realize the interpretation cycle for a given classification task, a series of issues must be addressed.

Definition of a Flexible Representation for the Event
The selection of the representation is one of the most challenging design decisions in that it impacts the entire process. The representation defines the vocabulary that will be used by the system to characterize observed events. We need the representation to provide a rich feature space that allows the system to discriminate between a wide range of events. Since we do not necessarily know the spectrum of events the system will encounter, it may be very difficult to specify a universal representation. At the same time, we must keep in mind that our objective is to perform surveillance in real-time on a limited computational budget. Therefore we must select a representation that strikes a balance between these two objectives.

Specification of an Efficient Process for Real-time Classification and Confidence Assessment
Within the pattern recognition community, a myriad of techniques exist for classification and confidence assessment. Unfortunately, many of these processes are not suitable for real-time applications due to their complexity. Careful consideration must be given to the design of the classification process in order to satisfy our objectives of reliably classifying known events and rejecting unknown events in real-time.

Design of a Process for Efficient Incremental Learning through Interaction with the User
Given the amount of data that a distributed video surveillance system with a large number of sensors will produce, the user requires techniques to efficiently identify the salient examples which require user interpretation. Our goal is to minimize the number of system queries required to obtain a classifier that meets the user's performance requirements.

1.5 Overview of Related Work in Video Surveillance

The set of challenges outlined above span several domains of research. We will review the majority of relevant work in future chapters where the research is considered in the context of the specific problem at hand. In this section, we will focus on the video surveillance systems discussed in the literature to understand the approaches pursued for various surveillance missions.

Due to the functional similarities between the video surveillance systems discussed in the literature, we begin by defining a general decomposition of the video interpretation process so that we have a framework for comparing system capabilities. Table 1.1 decomposes the video interpretation process into a set of *task-independent* and *task-dependent processes*.[1] We define a task-independent process as one whose definition is independent of the specific objectives of the surveillance mission. Low-level processes which derive features from the video data are generally task-independent processes. The definition of a task-dependent process, on the other hand, is dependent on the objectives of the surveillance mission. These processes are responsible for mapping features derived from the video into event categories that are meaningful and relevant to the user. Table 1.2 highlights which processes are employed in the various video surveillance systems considered.

For most of the systems listed in table 1.2, the common objectives are to detect, classify, track and localize objects of interest in the environment. Many of the systems also classify activity by analyzing object actions and object interactions. The global objective for the majority of these systems is to employ context specified *prior to deployment* to interpret activity in real-time and present a clear picture of the activity to the user. No interaction with the user takes place to discover new object classes and incrementally learn from the observations.

The *VIEWS* system [17, 12] is the earliest video surveillance system listed that attempts to describe activity in the scene using a model-based approach. VIEWS was designed to operate in environments where almost all of the possible situations can be specified in advance. The user provides camera models, a ground plane representation denoting the salient regions of the environment, 3-D object models and behavior models. Using this knowledge base, the system develops strategies offline to optimize the performance of the system. During operation, VIEWS requires significant computational power to achieve real-time performance. The objective of the *PASSWORDS* project [7, 16] was to provide similar functionality to VIEWS using low cost parallel digital signal processors.

One of the most ambitious video surveillance programs was the *Video Surveillance and Monitoring* (*VSAM*) program [47] led by Carnegie Mellon University and Sarnoff Corporation. The objective of VSAM was to demonstrate efficient wide-area video surveillance using a distributed network of electro-optical sensors. The CMU/Sarnoff system provided the capability to detect, classify, track, localize and visualize objects within the known environment. Calibrated cameras and a 3-D site model were used to localize objects efficiently by ray projection. Moving objects were classified into the categories of human, human group and vehicle. Motion of individual humans was classified into the categories of running and walking. Using the object location and class label information derived by the system, object models were inserted into the site model to provide the user with a global view of activity around the site.

Other video surveillance systems demonstrated under the VSAM effort included those developed by MIT [31, 46] and Texas Instruments [25]. The MIT philosophy for video surveillance deviates strongly from the typical approach in that they believe much of the context necessary to analyze activity in a site can be derived from observation. Using

[1] The definitions of movement and activity classification follow from the decomposition of the motion interpretation task proposed by Bobick [6].

Task-Independent Processes	*Definition*
Change Detection	Identification of regions in a given video frame that deviate significantly from the expected intensity values
Region Localization	Transformation of the 2-D image coordinates of the center of an image region into the corresponding 3-D world coordinates
Region Tracking	Association of image regions nominated by the change detector in consecutive video frames
Task-Dependent Processes	*Definition*
Object Tracking	Association of image regions nominated by the change detector in consecutive video frames that match the object model
Region Classification	Classification of the image regions nominated by the change detector
Movement Classification	Classification of consistent, predictable object motions
Activity Classification	Classification of statistical sequences of movements

Table 1.1: A general decomposition of the video interpretation process

robust change detection and region tracking procedures, Grimson et al. [31] demonstrate the capability to self-calibrate a distributed network of sensors and construct rough site models using motion cooccurrence to establish correspondence between the sensors. Object and activity classification are achieved at a low level by clustering detections and tracks with similar attributes. Invanov et al. [46] add a more sophisticated process for activity classification to this system that uses a stochastic context-free grammar to recognize a set of user defined events. The Texas Instruments system [25] is designed specifically to monitor human activities within and around office buildings. The system detects and groups regions nominated by the change detector into collections that are consistent with the size and shape of a person. The detected people are localized and tracked and their track information is stored in an object-oriented database termed the *visual memory*. The user can query the visual memory to focus on various activities of interest and event alarms can be interactively defined.

System	*Change Detection*	*Region Localization*	*Region Tracking*	*Object Tracking*	*Region Classification*	*Movement Classification*	*Activity Classification*
VIEWS	√	√	√		√		√
PASSWORDS	√	√	√		√		√
VSAM/CMU Sarnoff	√	√	√		√	√	
VSAM/MIT	√	√	√		√		√
Texas Instruments	√	√	√		√		√
Orwell et al.	√	√	√				
Remagnino et al.	√	√	√		√	√	√
Foresti	√	√	√		√		
W^4/W^4_{AT}	√			√	√		
W^4S	√	√		√	√		
CMU Cyberscout	√		√	√	√		

Table 1.2: Video surveillance system capabilities

Orwell et al. [57] define an agent-based architecture for video surveillance to simplify the process of interpreting multiple video streams and integrating the results into a coherent description of the scene. Within their system, there are two types of agents: camera agents and object agents. Camera agents are responsible for detecting and tracking moving objects in the video stream, updating the background model and creating object agents for stable objects. Object agents are responsible for updating the estimate of the object trajectory and communicating with other object agents to determine if there are multiple object agents for the same object that should be merged. Remagnino et al. [63] present another agent-based framework for automatically generating textual descriptions of activity in the scene. In this system, behavior agents reason about the actions of individual objects while situation agents assess the interactions among objects that are in close proximity. Bayesian networks are employed to interpret the activity.

Foresti describes two video surveillance systems that detect, classify, track and localize five object types in real-time. In [26], the *statistical pecstrum* ("specstrum") is used to obtain a representation of the object shape that is invariant to translations, rotations and scalings. In [27], the specstrum is replaced by the statistical morphological skeleton in order to reduce computational demands by employing a common representation for tracking and classification. The object signatures used to design the classifier in [27] are derived from 3-D object models specified by the user.

Haritaoglu et al. discuss several variations of the W^4 system in the literature that are designed to detect people and track their body parts. The original system [35] tracks the head, torso, arms and legs of upright people in real-time. W^4S [36] integrates the SRI stereo vision module with W^4 to overcome problems caused by sudden illumination changes, shadows and occlusions. W^4_{AT} [37] adds a pan-tilt-zoom camera along with the capability to detect moving objects while the camera is moving. Several other additions to W^4 are also presented that allow the system to track body parts in arbitrary postures [34], detect and track groups of people [38] and detect and track objects carried by people [33].

1.6 Objectives of the CMU Cyberscout Program

In this thesis, we will be discussing the design of the interpretation cycle for the CMU Cyberscout distributed surveillance system. Cyberscout consists of a collection of mobile and stationary sensor systems designed to detect, classify and track moving objects in the environment in real-time. Each sensor system consists of one or several electro-optical cameras with associated desktop PCs to process the video streams, and wireless communications links to exchange information with other sensor systems. The Cyberscout system is envisioned to operate in remote areas where prior knowledge of the environment is limited. Therefore we require the capability to obtain additional context from the user when necessary in order to identify relevant moving objects in the scene.

Given that the sensor systems will be constrained in terms of available power and computation, efficiency in the interpretation cycle is of the utmost importance. In the Cyberscout program, our goal is to demonstrate collaboration at various levels in the system to simplify the interpretation. At the lowest level, collaboration occurs between the various interpretation processes running on a given sensor system. Collaboration will also occur between processes running on different sensor systems. The final level of collaboration occurs between the system and the user.

Within the context of this thesis, collaboration will simply imply information exchange. Yet collaboration can involve the coordination of sensing, computation and communication resources distributed across the collection of sensor systems [23]. Since the necessary in-

frastructure does not currently exist to perform multi-sensor collaborative classification, we will be focusing on collaboration between the user and a single sensor system.

1.7 Thesis Overview

This thesis can be decomposed into two major sections: theory and experiments. In the next two chapters, we define the elements of the classification process and the principles for realizing the process. Chapter 2 provides an overview of the design. Chapter 3 addresses the details of learning to classify the image sequences from the data.

In the following three chapters, we investigate whether the classification process supports effective image sequence classification, novel image sequence detection and incremental learning in the context of a relevant surveillance task. Chapter 4 addresses the design and evaluation of the image classifier. Chapter 5 examines the design of the image sequence classifier and the classification and novelty detection performance of the entire process. Chapter 6 considers the relevance of the novelty detection process for efficient incremental learning.

Chapter 7 addresses the implementation of the classification process in the CMU Cyberscout system. Chapter 8 presents concluding remarks, contributions and avenues for future research.

Chapter 2

Classification Process Design

2.1 Overview

In this chapter, we will discuss the design of the classification process. We begin by specifying the general design requirements for the process. Then we discuss the design philosophy we believe will allow us to achieve these goals. Finally, we begin defining the elements of the classification process.

2.2 The Perception Processes

On each sensor system within the Cyberscout distributed video surveillance system, three processes interpret the video streams. The change detection process compares the incoming video frames with an adaptive background model and nominates regions with significant intensity change. The region tracking process attempts to match the candidate motion regions with regions from previous frames. Region tracking processes running on different sensor systems also communicate to determine if there are multiple regions being tracked that correspond to the same object. Example image sequences produced by the change detection and region tracking processes are shown in figure 2.1. The classification process labels a given image sequence as one of several object classes and assesses its confidence in the classification. The classification process also exchanges information with other classification processes in order to synthesize global decisions that are based upon all observations of the objects in the environment.

2.3 The Design Philosophy

In this thesis, we will focus on designing the classifier responsible for efficiently labeling the image sequences collected by a given sensor system. Our objective will be to simultaneously optimize the following parameters.

- *Computational Complexity*
- *Classification Performance*
- *Rejection Performance*

The issue of computational complexity will be the dominant concern in the design. Given that our goal is to deliver information about ongoing activity to the user in real-time, complex processes are not acceptable. We will strive to optimize the classification and

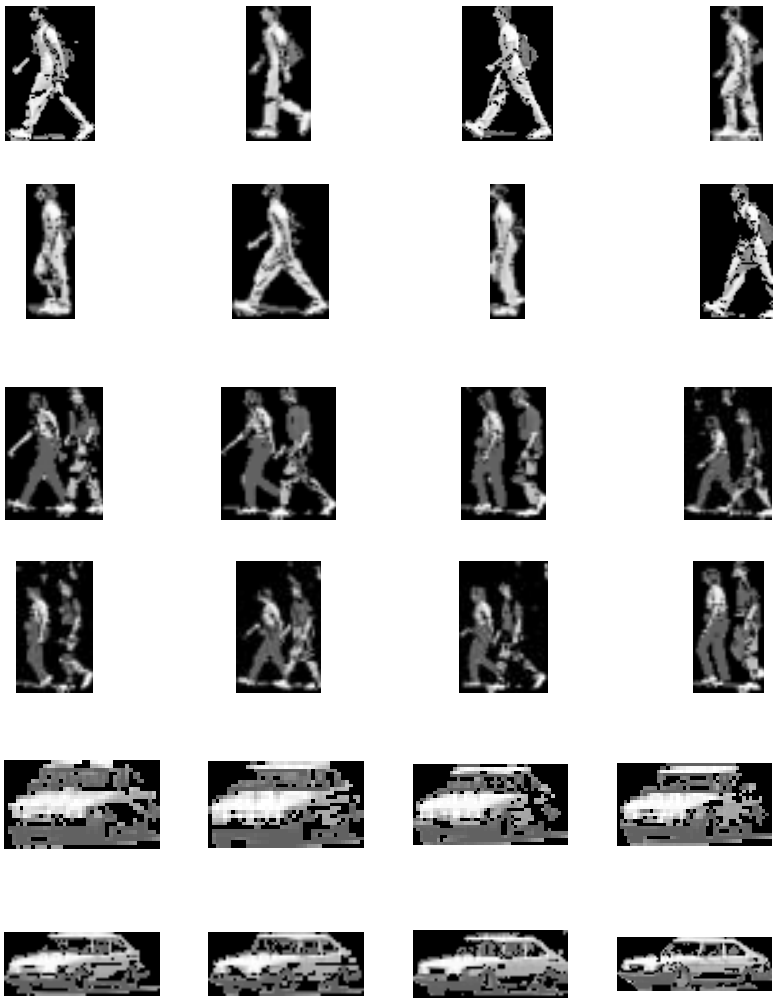


Figure 2.1: Example image sequences

rejection performance using the limited computational budget available. Simplicity will be stressed whenever possible.

Our approach to minimizing the burden of interpretation focuses on reconsidering how the sensing and interpretation processes should interact. Often algorithm designers assume that a classifier is provided with only one image to assess the nature of a particular object. Therefore they may attempt to obtain robustness to a variety of possible distortions by utilizing elaborate processes which are computationally expensive. Yet such classification processes are not well matched to the sensing process.

Surveillance systems generally sense the environment on a continuous basis. Therefore a classification decision need not be made after each sensor collection. If a particular collection yields an image that is ambiguous, there is no need to spend a significant amount of computation attempting to compensate for distortions which may have caused the ambiguity. Using a low complexity classifier with a measure of classification confidence, the system can resolve ambiguity by collecting imagery until a classification decision with an appropriate level of confidence can be made. Such a classification process is ideal for a distributed video surveillance system since multiple sensors can focus on an object in the environment from different locations, thereby increasing the likelihood of obtaining discriminating views during a particular collection. Through such a process, we believe we will be able to mitigate

the effects of malicious or ambiguous data while minimizing the computational demands on the sensing systems.

In order to specify the image sequence classification process, the following components must be defined: the representation of the image sequence, the classifier's decision process and the procedure for learning the classifier from the data. We consider each of these components next.

2.4 The Image Sequence Representation

The role of the representation is to provide a description of the image sequence that captures the necessary information for classification in a form that simplifies the process. The spectrum of possible representations can be partitioned into two types: *model-based* and *appearance-based representations*. A model-based representation consists of a user-defined parameterized model of a given object class and a procedure for estimating the model parameters from the image sequence. An appearance-based representation is a set of features derived from the image sequence through some transformation of the data. Model-based representations are limited to domains where the object classes have well defined geometric structures that can be easily parameterized. Appearance-based representations do not have this limitation which makes them applicable to a larger set of classification problems. Since we would like the user to be able to define new object classes simply by providing labeled image sequences, we will pursue an appearance-based representation.

When defining an appearance-based representation, the challenge is to specify a representation that is relatively insensitive to variations in the image data caused by changing environmental conditions and object-sensor configurations. Examining the image sequence examples in figure 2.1, one can observe several sources of variability that we must contend with. Images within a given sequence vary in size due to changes in object aspect and shape. Images also vary in resolution as the range from the object to the sensor changes. Object positions within the image are offset when partial detections or other image variations occur. Other challenges are caused by occlusion, lighting variation and nonuniform sampling of the environment.

In order to avoid the ill effects of these processes, there are two strategies one can pursue in the design of the representation. One option is to simply ignore features of the image sequence that the designer believes will not provide consistent, relevant information for classification. [1] Another option is to normalize the representation in a manner that removes or reduces the unwanted variability. We will employ both strategies in the definition of our representation.

When assessing the value of a given feature to the task, we must also consider the computational cost relative to the potential gain offered. Let us consider the value of the spatiotemporal information provided by the image sequences. In certain scenarios, the variation in appearance of the object over time can provide additional features for discriminating between object classes. Such features can be especially valuable in cases where only low resolution video of the object is available [21]. Yet in order to exploit this information, significant effort will be required to compensate for the malicious effects caused by changes in the environment, occlusion and nonuniform sampling. At the same time, the data requirements for acceptable generalization performance will be substantial since the system must learn to classify appearance variations at various resolutions and aspects.

[1] Prior to designing the classifier, the assessment of a feature's relevance to the task can not be conducted in a systematic manner. We must rely initially on the experience of the designer to guide the selection of the representation. Following the design, the relevance of the specified features can be assessed rigorously when a differentiable classifier is used [20, 39].

Given that in the majority of cases, the spatial features will provide enough information for classification, we will avoid exploiting the spatiotemporal features due to the anticipated adverse effects on system performance.

By limiting our focus to spatial features, the representation design task becomes one of defining an image representation. Over the years, countless representations have been proposed for characterizing the shape and intensity variation in image regions. Generally the goal is to define some low dimensional representation of the image that reduces the complexity of the classification task. By performing dimensionality reduction prior to designing the classifier, one does not know if information that is relevant to the classification task is being eliminated. Therefore we would like to limit the transformations of the data performed prior to designing the classifier and learn a feature set that preserves the necessary information for classification.

In cases where the set of possible object classes is unknown *a priori*, the representation should provide a rich description that allows the system to discriminate between the known object classes and other unknown objects. In recent work on appearance-based object detection and classification in static imagery, excellent performance has been achieved by classifying the images directly or features derived from the images through information-preserving transforms. Schneiderman [72] constructed detectors for faces and cars by modeling the class-conditional probability densities for a wavelet coefficient representation of the images. Papageorgiou [58, 60, 59] trained support vector machines to detect pedestrians and faces using wavelet coefficients derived from an overcomplete Haar wavelet dictionary. Roobaert [65, 66] trained linear support vector machines to successfully discriminate between a large number of object classes using the images directly.

Since we are not faced with the challenge of classifying objects embedded in complex backgrounds as in [72, 58, 60, 59], we will work directly with the images. In order to obtain robustness to scale variations, we will resize the images so that the largest dimension is a fixed dimension N. Then we will zero pad the original image to construct a square $N \times N$ pixel image with the original image in the center. By using this representation, we are not attempting to counter scalings or translations caused by partial detections or image variations that are included in an image region. Our assumption is that such events are transients that will not persist throughout the entire image sequence. Therefore robustness to such distortions will come from continuous sensing from multiple perspectives. In a later chapter, we will reconsider this topic in further detail when discussing the issue of image normalization.

2.5 Image Sequence Classification

Now that we have defined an initial representation, we will address the problems of learning to classify the size-normalized image sequences and assigning confidence levels to the classifications. As we discussed in the previous section, the classification process will consider only the spatial features of each image in order to determine the appropriate class label for the image sequence. The order of the images in the sequence will not influence the classification process. Under this constraint, the classification of an image sequence becomes a two-step process. In the image classification phase, a given image is analyzed to determine the likelihoods of the various classes. Then in the sequence classification phase, the evidence from the classification of the current image is integrated with past evidence to produce an overall decision with a confidence level. In this section, we will consider two approaches for learning an approximation to the Bayes-optimal image sequence classifier.

2.5.1 The Probabilistic Approach

In order to classify image sequences and estimate classification confidence, a natural approach is to estimate the *a posteriori* class probabilities $\mathrm{P}(\omega_k|S)$. The Bayes-optimal class label ω_* for the image sequence $S = \{S^1, S^2, \ldots, S^M\}$ is defined as

$$\omega_* = \underset{\omega \in \{\omega_1, \omega_2, \ldots, \omega_\mathcal{C}\}}{\operatorname{argmax}} \mathrm{P}\left(\omega | S^1, S^2, \ldots, S^M\right) \tag{2.1}$$

and classification confidence is generally expressed in terms of the probability of correct classification $\mathrm{P}_{CC|S} = \mathrm{P}\left(\omega_* | S^1, S^2, \ldots, S^M\right)$. According to Bayes' rule, the *a posteriori* class probabilities $\mathrm{P}\left(\omega_k | S^1, S^2, \ldots, S^M\right)$ can be expressed as

$$\begin{aligned} \mathrm{P}\left(\omega_k | S^1, S^2, \ldots, S^M\right) &= \frac{\rho\left(S^1, S^2, \ldots, S^M | \omega_k\right) \mathrm{P}(\omega_k)}{\rho\left(S^1, S^2, \ldots, S^M\right)} & (2.2) \\ &= \frac{\rho\left(S^1, S^2, \ldots, S^M | \omega_k\right) \mathrm{P}(\omega_k)}{\sum_{c=1}^{\mathcal{C}} \rho\left(S^1, S^2, \ldots, S^M | \omega_c\right) \mathrm{P}(\omega_c)}. & (2.3) \end{aligned}$$

Therefore in order to specify the Bayes-optimal classifier, the class prior probabilities $\mathrm{P}(\omega_k)$ and the class-conditional image densities $\rho\left(S^1, S^2, \ldots, S^M | \omega_k\right)$ must be defined. To simplify the definition of the densities, we enforce the constraint that the classifier should be invariant to the ordering of the images. In order for the invariance to hold, we will assume the following form for the class-conditional image densities

$$\rho\left(S^1, S^2, \ldots, S^M | \omega_k\right) = \prod_{i=1}^{M} \rho\left(S^i | \omega_k\right) \tag{2.4}$$

which implies the images in the sequence are independent. By imposing this independence assumption, the partitioning of the image sequence classification process becomes clear. Classifying the individual images amounts to evaluating the class-conditional image densities $\rho\left(S^i | \omega_k\right)$ for all possible object classes. Then classifying the sequence involves simply applying Bayes' rule to integrate the evidence and determine the most likely object class. Obviously the assumption of independence is far from reality, so one may be concerned about the performance of such a classifier. Yet let us suspend our disbelief for a moment and probe deeper.

Consider the problem of learning the class-conditional image densities from the labeled image sequences. The standard assumption invoked in such a task is that the labeled examples result from independent, identically distributed trials. Clearly the images in a given sequence fail to satisfy this assumption. If we attempt to learn the densities from the set of image sequences, the density of images throughout image space would be related to the lengths of the training sequences, which are determined by the movement of the objects in the scene and the field of view of the sensor. This implies that we would have little hope of the estimated densities converging to meaningful distributions. Therefore we need to investigate another approach.

2.5.2 Partitioning Image Space

Let us step back for a moment and consider the classification problem in general terms. Classification involves mapping feature vectors in a given feature space to one of several

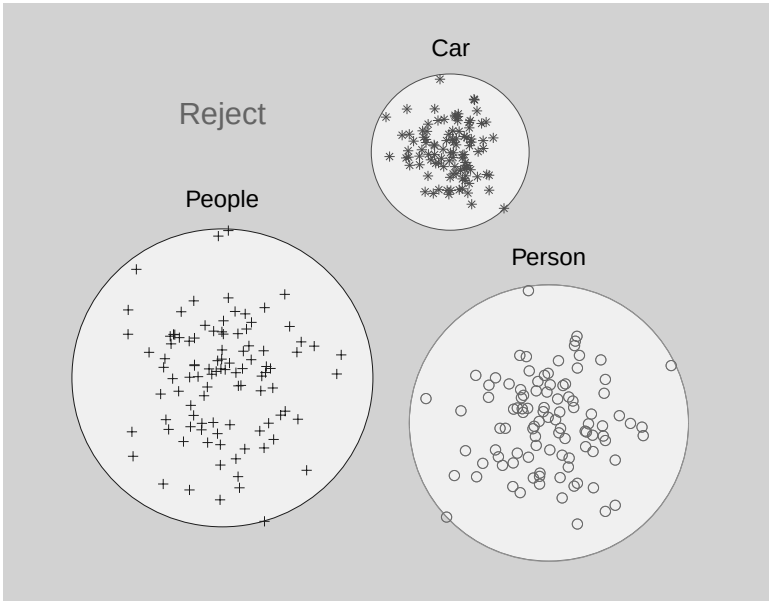


Figure 2.2: A sample partition

class labels. Given a set of feature vectors along with their corresponding class labels, the goal is to learn an approximation to the mapping represented by the training data. In most real-world problems, the actual mapping is not deterministic. A given feature vector is mapped stochastically to one of several class labels. Therefore one attempts to learn a deterministic mapping with the minimum expected value of a given risk measure. The most common risk measure employed in classification problems is the probability of error which leads to the Bayes-optimal classification procedure.

Learning an approximation to the mapping represented by the training data can be thought of as learning a partition of the feature space into a set of $\mathcal{C}$ decision regions. Each decision region R_k is defined as the collection of all points in the feature space that map to the class ω_k. In cases where the classifier is allowed to reject feature vectors, the partition will contain an additional decision region R_{reject} containing all feature vectors that can not be classified with an acceptable level of risk.

By learning approximations to the class-conditional image densities, we are learning an approximation to the Bayes-optimal partition indirectly. With few exceptions, the quality of this approximation will be less than ideal since the objective is to approximate the densities instead of the decision regions [44, 22]. Therefore one may question the motivation for estimating the densities. If the approximation to the Bayes-optimal partition is poor, the rejection performance for the approximate partition will likely be far from optimal as well.

So instead of attempting to learn class-conditional image densities, our objective will be to learn a partition of the image space directly with a low probability of error. Figure 2.2 illustrates our objective graphically. Given a set of training images and their corresponding class labels, we want to learn a partition that maps regions of image space to one of the specified object classes where significant data exists to support the decision. In other regions of image space where little to no data exists, feature vectors will be rejected. Given that the classifier will never have complete knowledge of the objects in the environment, the rejection capability will be important for the detection of unknown objects and novel views of known objects.

2.5.3 Classifying Class Label Sequences

As the image classifier processes a series of images of a given object, a sequence of class labels is produced. Based on this class label sequence, we wish to assign a class label to the *image sequence* along with a level of classification confidence. The class label sequence provides two types of information. First, a set of possible class labels is obtained along with the relative frequencies of occurrence of each class label. In addition, the sequence captures the transitions between the class labels as the appearance of the object changes over time. The question we wish to address at this point is whether the class label distribution is sufficient to reliably classify image sequences of known objects and detect image sequences of unknown objects and novel views of known objects.

Whether or not the class label distribution is sufficient is actually determined by the image representation and the partition. In order to reliably classify known objects and detect unknown objects and novel views of known objects based on the class label distribution, we must be able to successfully discriminate between such examples in image space. If the combination of image representation and partition provides the necessary discrimination power, the class label distributions induced by known objects, unknown objects and novel views of known objects will be sufficiently separable in class label distribution space. Ideally one would hope that the image classifier reliably and consistently classifies or rejects the images in a given sequence as illustrated in figures 2.3(a) and (b), thereby simplifying the image sequence classification task. Yet the reality is that image sequences will often induce a mixture of classifier outputs as illustrated in figure 2.3(c) indicating classifier confusion. When classifier confusion does occur, our ability to discriminate between known and unknown objects does not necessarily decrease significantly. As we shall see later, certain unknown object classes may actually display patterns of classifier confusion that are different from those associated with the known object classes. Therefore the task of identifying the unknown object image sequences remains tractable.

Since it is not clear what additional patterns could be efficiently exploited in the history of the class label transitions, our approach to sequence classification will entail mapping class label distributions to one of the known object classes. One possible solution involves simply selecting the class label that occurs most frequently and using the fraction of labels corresponding to the most frequent class label as a measure of confidence [52]. Although we expect such a rule will generally work well for classification, the measure of confidence is suspect due to the dependencies in the image sequence. At the same time, the type of classifier confusion is ignored in the assessment of confidence. Instead of using this heuristic, we will learn a partition of the class label distribution space from the training data and classification confidence will be assessed in a more principled manner. We will withhold our discussion of classification confidence until later, since our approach is connected with the learning procedure.

2.6 Conclusions

When the available computational resources are constrained, we must think carefully about leveraging the capabilities of the surveillance system to maximize the performance of the classification process. In this chapter, we have argued that the ability to continuously observe moving objects over time from a variety of perspectives is the key to meeting our performance objectives. Instead of using a complex process to resolve ambiguity in individual images, we will employ a low complexity process to classify objects over time as they are observed.

The overall classification process will be decomposed into two classification tasks: image classification and class label distribution classification. The image classifier will label

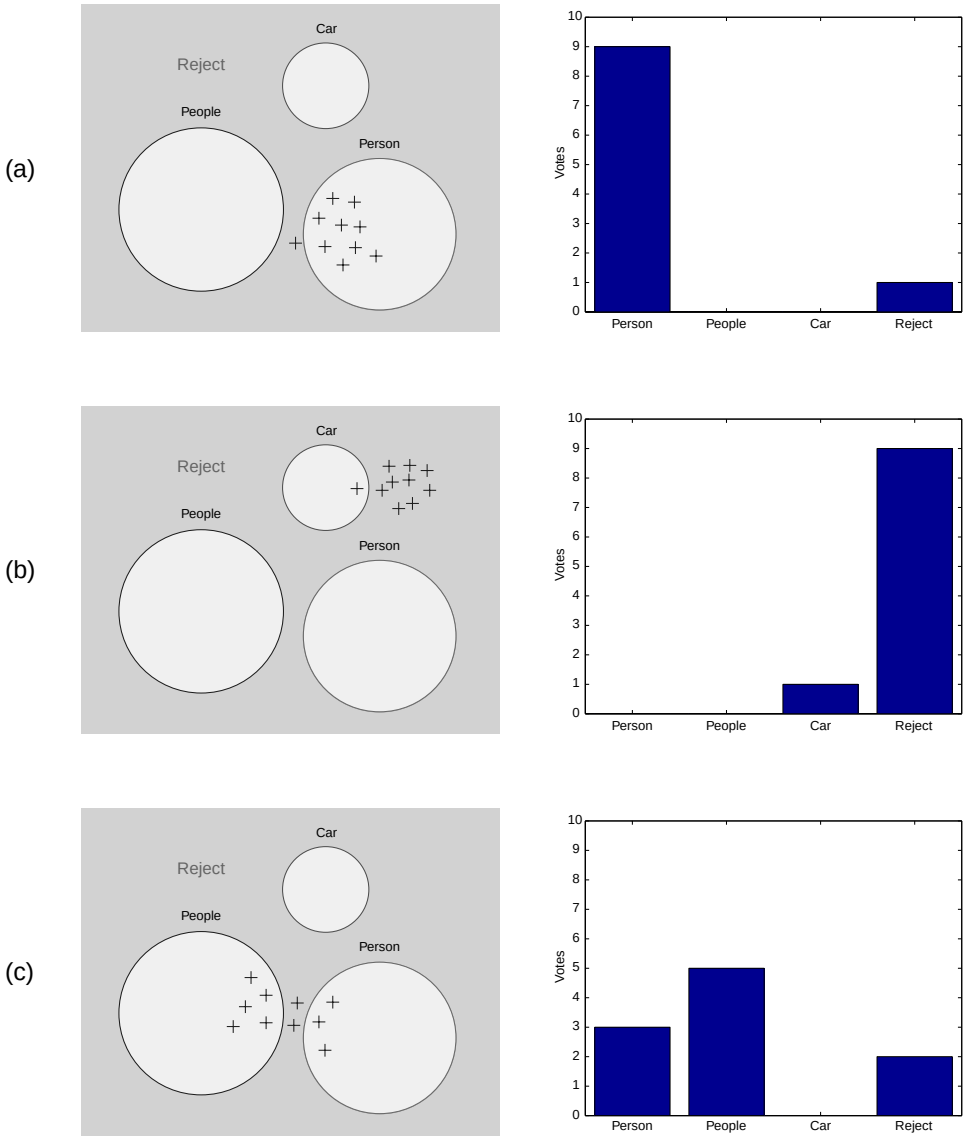


Figure 2.3: Examples of image sequence class label distributions: (a) confident classification: vast majority of the images lie in one decision region (b) consistent rejection: significant fraction of the images lie in the rejection region (c) classifier confusion: images distributed over two or more regions

size-normalized images in order to support classification of known objects and rejection of unknown objects and novel views of known objects. The class label distribution classifier will label image sequences and assess classification confidence based upon the type of class label variation. We will learn each of these partitions directly from the data using the learning strategy developed in the following chapter.

Chapter 3

Learning Theory

3.1 Overview

In order to learn the image and class label distribution classifiers, we must address the same general learning problem. In this chapter, we will present the mathematical definition of the problem. Then we will discuss the theory of large margin classification and its relevance to the problem. We will review the current techniques for large margin classification and evaluate the suitability of each method in the context of the design objectives. After selecting a technique to serve as the baseline learning algorithm, we will introduce several modifications to the learning procedure that are necessary to address specific challenges associated with the image sequence classification task. Finally, we will explore the issues of confidence assessment and rejection within the proposed classification process.

3.2 Learning a Partition of Feature Space

3.2.1 Learning Indicator Functions in Feature Space

For both classification tasks, our objective is to learn a partition of the given feature space directly with a low probability of error. This can be thought of as learning a set of $\mathcal{C}$ *indicator functions* $\mathcal{I}_{\omega_k}(X|\theta)$ in feature space which are defined as

$$\mathcal{I}_{\omega_k}(X|\theta) = \begin{cases} 1 & \text{if } X \in R_k(\theta) \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The general risk measure we would like to minimize with respect to the function parameters θ is

$$1 - \mathrm{E}_{\mathbf{X},\Omega}\left[\mathcal{I}_{\omega_X}(X|\theta)\right]. \tag{3.2}$$

In order to define the indicator functions, we must specify some parameterized representation of the partition of feature space. The general approach we will follow involves defining a set of $\mathcal{C}$ *discriminant functions* $g_k(X|\theta)$ where

$$g_k(X|\theta) - \max_{i,i\neq k} g_i(X|\theta) > 0 \tag{3.3}$$

when the example X maps to the class label ω_k [24]. The set of examples satisfying equation 3.3 define the decision region $R_k(\theta)$. In order to define the rejection region, examples

within a certain region about the decision boundaries will be rejected. For a given class ω_k, all examples that lie within the portion of the decision region R_k defined by

$$0 \leq g_k(X|\theta) - \max_{i,i\neq k} g_i(X|\theta) \leq \delta_{R_k} \tag{3.4}$$

will be rejected. Confidence in the classification is assumed to increase as the difference

$$g_k(X|\theta) - \max_{i,i\neq k} g_i(X|\theta) \tag{3.5}$$

becomes increasingly positive. Therefore in regions close to the decision boundary, where the difference between the largest and next largest discriminant function is small, the resulting classifications are assumed to be unreliable. Later in this chapter, we will revisit the issue of rejection in order to address this strategy in more detail.

Based on these definitions, the indicator function $\mathcal{I}_{\omega_k}(X|\theta)$ is defined as

$$\mathcal{I}_{\omega_k}(X|\theta) = \begin{cases} 1 & \text{if } g_k(X|\theta) - \max_{i,i\neq k} g_i(X|\theta) > \delta_{R_k} \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

To specify the indicator functions completely, we must first select a parametric functional form for the discriminant functions. This defines the set of candidate partitions of the feature space known as the *hypothesis class*. Then we must estimate the parameters θ and δ_{R_k} defining a partition that will generalize well to unseen examples.

3.2.2 Empirical Risk Minimization

When learning a classifier from a set of training examples, our general goal is to minimize the expected value of a given loss function $L(X, \omega, \theta)$ with respect to the classifier parameters θ. Typically this is achieved by minimizing the *empirical risk functional* [82]

$$R_{emp}(\theta) = \frac{1}{N} \sum_{j=1}^{N} L\left(X^j, \omega^j, \theta\right) \tag{3.7}$$

over the training data $\left\{\left(X^1, \omega^1\right), \left(X^2, \omega^2\right), \ldots, \left(X^N, \omega^N\right)\right\}$. Within the neural network community, the common loss functions employed include the squared error, Minkowski error and cross-entropy [5]. By employing these error measures, minimizing the empirical risk functional amounts to learning an approximation to the *a posteriori* class probabilities $\mathrm{P}(\omega_k|X)$ [42][44, Ch. 2]. In order to minimize the error rate on the training data directly, the empirical risk functional to minimize is

$$R_{emp}(\theta) = \frac{1}{N} \sum_{j=1}^{N} 1 - \mathcal{I}_{\omega^j}\left(X^j|\theta\right) \tag{3.8}$$

$$= 1 - \frac{1}{N} \sum_{j=1}^{N} \mathcal{I}_{\omega^j}\left(X^j|\theta\right). \tag{3.9}$$

Let us consider the merits of selecting a partition by minimizing the error rate over the training set, assuming for the moment that we can identify the global minima of the above risk functional. In the limit of infinite training data, the minimum of the empirical risk

functional converges to the minimum risk achievable with the specified hypothesis class if the hypothesis class satisfies the conditions detailed in [82, Ch. 2]. When a limited amount of training data is available, minimizing the empirical risk functional does not guarantee that the actual risk of the resulting partition is close to the minimum achievable risk. The success of empirical risk minimization is determined by the *capacity* of the hypothesis class [19] which refers to the richness of the set of possible partitions in the hypothesis class. As the capacity increases, the number of partitions in the hypothesis class that minimize the empirical risk functional increases, thereby increasing the likelihood of significant deviations between the training error and the generalization error. Therefore in order to achieve acceptable generalization performance, it will be imperative to control the capacity of the hypothesis class.

3.2.3 Bounding the Expected Risk

Vapnik and Chervonenkis developed a rigorous measure of the capacity of a set of indicator functions and a family of bounds characterizing the generalization performance of a classifier [82]. We will focus on the following bound on the *expected risk* $R(\theta)$ [32, 11] which states that with probability $1 - \eta$

$$R(\theta) \leq R_{emp}(\theta) + \varepsilon(N, h, \eta) \tag{3.10}$$

where

$$\varepsilon(N, h, \eta) = \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{N}} \tag{3.11}$$

and $N > h$. The bound on the deviation $\varepsilon(N, h, \eta)$ is a function of the number of training examples N, the confidence parameter η and the *Vapnik Chervonenkis* (VC) *dimension* h of the classifier. The VC dimension of a set of indicator functions $\mathcal{F} = \{\mathcal{I}(X|\theta)\}$ is the maximum number of points in the feature space $\mathbf{X}$ that can be *shattered* by the set $\mathcal{F}$. A set of points is shattered by the set $\mathcal{F}$ if all possible labelings of the points can be realized by $\mathcal{F}$. We refer the interested reader to [11, 19] to learn more about the VC dimension.

The bound in equation 3.10 rigorously illustrates the tradeoff that must be made in order to obtain a classifier that generalizes well when limited training data is available. As the capacity of the hypothesis class is increased, the empirical risk either remains constant or decreases. At the same time, the *VC Confidence* $\varepsilon(N, h, \eta)$ increases monotonically with increasing VC dimension. Therefore the designer must find a suitable balance between the empirical risk and the capacity in order to minimize the bound on the expected risk.

3.2.4 Controlling the Capacity

Typically several steps are taken to control the capacity of the hypothesis class. Before learning a classifier, designers often employ some technique for *dimensionality reduction* which involves transforming the training data into a lower complexity representation. By reducing the dimensionality of the training data, the number of parameters in the discriminant functions is reduced. The designer may also adjust the complexity of the discriminant functions by *modifying the structure*. This can amount to varying the number of kernels in a radial basis function network or the number of hidden units in a multi-layer perceptron. Finally, if the capacity must be further constrained during the learning process, *regularization* techniques such as weight decay can be used to penalize overly complex partitions of the feature space. In general, the classical approach has been to *minimize the number*

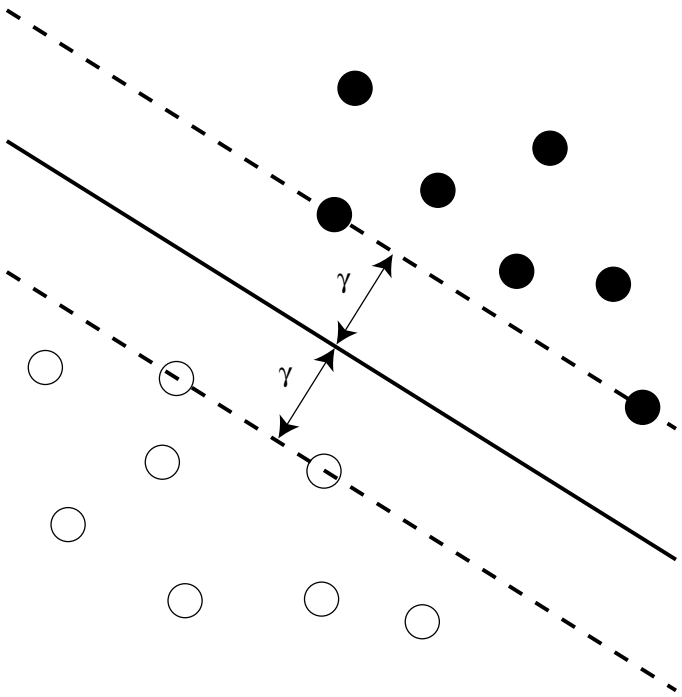


Figure 3.1: Margin of a linearly separable training set

of parameters without excessively hindering performance on the training data in order to obtain a partition that generalizes well.

This approach is entirely consistent with the approach suggested by VC theory. Yet within the last decade, a new class of learning procedures has emerged that seemingly contradicts VC theory while providing state-of-the-art classification performance on a broad spectrum of problems. The methods are referred to as techniques for *large margin classification*. In the following sections, we will consider general properties of these methods along with specific approaches for learning large margin classifiers. We will assess the suitability of these methods for learning partitions that satisfy our requirements.

3.3 Large Margin Classification

3.3.1 Rosenblatt's Perceptron

The roots of large margin classification trace back to the earliest work in statistical learning theory that focused on the first learning machine introduced by Rosenblatt: *the perceptron*. The perceptron is a thresholded linear machine $g(X|\theta, \theta_b)$

$$g(X|\theta, \theta_b) = \text{sgn}\left(\theta \cdot X + \theta_b\right) \tag{3.12}$$

defined by the parameters (θ, θ_b) and trained online by incrementally adjusting the parameters as the machine misclassifies the training examples [67]. The perceptron partitions a given feature space $\mathbf{X}$ into two half spaces using a hyperplane and is guaranteed to find a partition that classifies the training data without error if the training set is linearly separable.

Soon after the perceptron's introduction in the early sixties, Novikoff [56] proved the following result about the perceptron. Let $R = \max_i \|X^i\|$. Suppose γ is the largest real

number satisfying the condition

$$Y^i\left(\theta{\cdot}X^i+\theta_b\right)\geq\gamma>0 \tag{3.13}$$

for all training examples $\left\{\left(X^1,Y^1\right),\left(X^2,Y^2\right),\ldots,\left(X^N,Y^N\right)\right\}$ when $\|\theta\|=1$ and $Y^i\in\{-1,1\}$. Then the number of mistakes made by the perceptron learning rule on the training set is at most

$$\left(\frac{2R}{\gamma}\right)^2. \tag{3.14}$$

γ is referred to as the *margin of the training set*. R is the radius of the smallest sphere that encloses all of the training examples.

To understand the concept of the margin, let us consider the geometric interpretation of the left hand side of equation 3.13. Given

$$Y^i\left(\theta{\cdot}X^i+\theta_b\right)>0, \tag{3.15}$$

we know

$$\begin{aligned} Y^i\left(\theta{\cdot}X^i+\theta_b\right) &= \left|Y^i\left(\theta{\cdot}X^i+\theta_b\right)\right| &(3.16)\\ &= \left|\theta{\cdot}X^i+\theta_b\right| &(3.17)\end{aligned}$$

since $Y^i\in\{-1,1\}$. Let us define the vector X_0 that is parallel or antiparallel to θ and lies on the decision boundary defined by the hyperplane

$$\theta{\cdot}X_0+\theta_b=0. \tag{3.18}$$

Substituting for θ_b, we find

$$Y^i\left(\theta{\cdot}X^i+\theta_b\right)=\left|\theta{\cdot}(X^i-X_0)\right|. \tag{3.19}$$

When $\|\theta\|=1$, $\left|\theta{\cdot}(X^i-X_0)\right|$ equals the distance from the training example X^i to the decision boundary. This implies that if equation 3.13 is satisfied for all training examples, the training data can be separated by a hyperplane such that all training examples are at least a minimum distance γ from the hyperplane. The quantity $\left(\frac{R}{\gamma}\right)^2$ therefore provides a measure of maliciousness by comparing the spread of the training set in the feature space to the maximum separation achievable between the classes. As we shall see, this is a measure that has reappeared in recent work attempting to explain the excellent generalization performance of large margin classifiers.

3.3.2 Maximizing the Margin

As stated in the previous section, the perceptron learning procedure is guaranteed to produce a hyperplane that classifies the training set without error if the training set is linearly separable. Since there are typically numerous partitions that achieve error free classification when the training set is linearly separable, the perceptron learning procedure will not produce a unique solution. Some additional risk measure is needed to rank the partitions that minimize the error rate on the training set. One strategy is to maximize the distance between the hyperplane and the training examples that are closest to the hyperplane. The

hyperplane that achieves the maximum separation is referred to as the *maximal margin hyperplane*.

Let us return to the risk bound from VC theory in order to analyze the generalization performance of the maximal margin classifier. Recall that the expected risk $R(\theta)$ is bounded by

$$R_{emp}(\theta) + \varepsilon(N, h, \eta) \tag{3.20}$$

with probability $1 - \eta$ where

$$\varepsilon(N, h, \eta) = \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{N}}. \tag{3.21}$$

Given the maximal margin classifier separates a linearly separable training set without error, $R_{emp}(\theta) = 0$. Therefore the expected risk is bounded solely by the VC confidence $\varepsilon(N, h, \eta)$.

The only remaining quantity to specify is the VC dimension h. As we discussed previously, the VC dimension is a measure of capacity that refers to the maximum number of points that can be shattered by the set of indicator functions. For the set $\mathcal{F}$ of all possible hyperplanes in $\mathbb{R}^m$, the maximum number of points that can be shattered is $m + 1$ [24, 11]. Therefore the risk bound for a hyperplane that partitions the training set without error is

$$R(\theta) \leq \sqrt{\frac{(m+1)\left(\log\left(\frac{2N}{m+1}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{N}} \tag{3.22}$$

for $N > m + 1$. As one would expect, the bound states that as the dimensionality of the feature space increases, the number of training examples must increase proportionately if we are to have any hope of generalizing to unseen examples. At the same time, it suggests that as the dimensionality of the feature space tends toward infinity, we will never have enough data to generalize well with high probability.

Now let us consider the VC dimension of the set of hyperplanes $\mathcal{G}$ that achieve a margin γ in $\mathbb{R}^m$. Due to the additional constraint placed on the set of admissible partitions, we expect that the VC dimension of the set $\mathcal{G}$ is smaller than the VC dimension of the set $\mathcal{F}$. In fact, Vapnik [81] has shown that the VC dimension for the set of hyperplanes that achieve a margin γ in $\mathbb{R}^m$ is bounded by

$$h \leq \min\left(\left(\frac{R}{\gamma}\right)^2, m\right) + 1. \tag{3.23}$$

This is a remarkable result in that it suggests the maximal margin classifier has the potential to generalize well even in infinite dimensional spaces if the distribution of the data is benign. Notice once again that the measure $\left(\frac{R}{\gamma}\right)^2$ is intimately connected with the generalization performance of the learning procedure. This quantity is an upper bound on the *scale-sensitive VC dimension* (*fat shattering dimension*) which is defined as the largest number of points that can be shattered by a set of indicator functions with margin γ [19].

In the last few years, a number of other *data-dependent generalization bounds* based on various measures of the margin have been presented [19] that are *independent of the dimensionality of the feature space*. These bounds are very intriguing in that they suggest it is possible to avoid the curse of dimensionality without reducing the number of parameters in the discriminant functions. By maximizing the margin of the classifier in the feature

space, the capacity of the hypothesis class can be effectively controlled. At the same time, the bounds also indicate that it may be possible to *improve generalization performance by increasing the dimensionality of the feature space*. Such a notion completely contradicts traditional approaches to classifier design. Yet a variety of experiments have demonstrated that increasing the dimensionality can actually lead to improvements in performance when the capacity is controlled through maximization of the margin [59, 60, 71].

The implication of these results is very important in the context of the image classification task. Traditionally, the definition of a low dimensional image representation has been an important step in the design of an image classifier since high dimensionality will generally guarantee poor generalization performance when limited training data is available and classical learning techniques are employed. Based on the studies of generalization that incorporate some measure of the complexity of the sample, we now see that dimensionality reduction is not a prerequisite to achieve improvements in generalization. With large margin classification techniques, it may be possible to effectively classify images of known objects in high dimensional feature spaces that preserve enough discriminant information to support the rejection of unknown objects and novel views of known objects.

The degree of our success will be determined by the complementary nature of the representation and the hypothesis class. In essence, we are in search of an effective combination of high dimensional representation and hypothesis class that admits a succinct description of the partition. The scale-sensitive VC dimension is one such measure that captures the complexity of the resulting description for linearly separable problems. Other margin-based measures have been used to establish bounds on the generalization performance when the training set cannot be classified without error, further establishing the theoretical underpinnings for the success of large margin classification [19].

3.4 Techniques for Large Margin Classification

In this section, we review three classes of large margin classification procedures: *support vector machines*, *boosting* and *differential learning*. Our goal is to explore each perspective, highlight the distinct aspects and common threads and assess the suitability of the procedures for our specific task.

3.4.1 Support Vector Machines

The support vector approaches to classification are based on the established theories of generalization that highlight the benefits in performance achieved through the maximization of margin-based measures. Support vector machines are simply large margin hyperplanes that partition either the original feature space or a higher dimensional space that the original feature space is embedded in. In this overview, we will first discuss the procedure for learning the maximal margin hyperplane for linearly separable problems. Then we will consider modifications that allow the learning procedure to address nonseparable problems.

3.4.1.1 Learning the Maximal Margin Hyperplane

In order to learn the maximal margin hyperplane, we must solve a constrained optimization problem. The optimization problem is stated as

$$\begin{gathered} \textit{minimize } \|\theta\|^2 \textit{ with respect to } (\theta, \theta_b) \\ \textit{subject to the constraints } Y^i \left(\theta \cdot X^i + \theta_b\right) \geq 1 \textit{ for } i \in \{1, \ldots, N\}. \end{gathered}$$

Since the hyperplane parameters can be uniformly scaled by a positive constant without changing the partition, the constraints are defined such that the magnitude of the classifier output must always be greater than or equal to a given constant which is 1 in this case. Identifying the hyperplane with minimum norm subject to these constraints is equivalent to maximizing the margin when $\|\theta\|$ is fixed.

This constrained convex minimization problem can be approached in two ways. The primal form of the problem involves minimizing the Lagrangian [11]

$$L_P(\theta, \theta_b, \boldsymbol{\alpha}) = \frac{1}{2}\|\theta\|^2 - \sum_{i=1}^{N} \alpha_i \left(Y^i \left(\theta \cdot X^i + \theta_b \right) - 1 \right) \tag{3.24}$$

with respect to (θ, θ_b) subject to the constraints

$$\frac{\partial L_P}{\partial \alpha_i} = 0, \quad \alpha_i \geq 0, \quad i \in \{1, \ldots, N\}. \tag{3.25}$$

The dual form of the problem involves *maximizing* L_P subject to the constraints

$$\frac{\partial L_P}{\partial \theta} = 0, \quad \frac{\partial L_P}{\partial \theta_b} = 0, \quad \alpha_i \geq 0 \quad i \in \{1, \ldots, N\}. \tag{3.26}$$

Differentiating L_P with respect to (θ, θ_b) and setting the results equal to zero, we find

$$\theta = \sum_{i=1}^{N} \alpha_i Y^i X^i \tag{3.27}$$

$$\sum_{i=1}^{N} \alpha_i Y^i = 0. \tag{3.28}$$

Substituting the equality constraints into equation 3.24, we obtain the dual Lagrangian [19]

$$\begin{aligned} L_D(\boldsymbol{\alpha}) &= \frac{1}{2}\|\theta\|^2 - \sum_{i=1}^{N} \alpha_i \left(Y^i \left(\theta \cdot X^i + \theta_b \right) - 1 \right) && (3.29) \\ &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} Y^i Y^j \alpha_i \alpha_j \left(X^i \cdot X^j \right) - \sum_{i=1}^{N} \sum_{j=1}^{N} Y^i Y^j \alpha_i \alpha_j \left(X^i \cdot X^j \right) \\ &\quad - \theta_b \sum_{i=1}^{N} \alpha_i Y^i + \sum_{i=1}^{N} \alpha_i && (3.30) \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} Y^i Y^j \alpha_i \alpha_j \left(X^i \cdot X^j \right). && (3.31) \end{aligned}$$

This allows us to restate the dual form of the problem as the maximization of the dual Lagrangian L_D with respect to the Lagrange multipliers $\boldsymbol{\alpha}$ subject to the constraints

$$\sum_{i=1}^{N} \alpha_i Y^i = 0, \ \alpha_i \geq 0 \quad i \in \{1, \ldots, N\}. \tag{3.32}$$

The solution to this optimization problem can be shown to satisfy the constraints [11]

$$\alpha_i \left(Y_i \left(\theta \cdot X_i + \theta_b \right) - 1 \right) = 0 \quad \forall \; i. \tag{3.33}$$

The implication of these constraints is that nonzero Lagrange multipliers are only associated with the training examples that satisfy the condition

$$Y_i \left(\theta \cdot X_i + \theta_b \right) = 1 \tag{3.34}$$

which indicates these examples are closest to the hyperplane. Combining this result with equation 3.27, we can see that the weight vector θ_* maximizing the margin is simply a linear combination of the training examples closest to the maximal margin hyperplane. The corresponding bias term θ_{b*} is determined by manipulating one of the above constraint equations.

The interesting aspect of this solution is the fact that the maximal margin hyperplane is uniquely determined by the subset of examples that are closest to the hyperplane. These training examples are referred to as the *support vectors of the training set*. The corresponding maximal margin hyperplane is referred to as a *linear support vector machine* (*SVM*). If we were to eliminate all of the examples from the training set other than the support vectors, the solution of the above optimization problem would still produce the linear SVM for the training set. Therefore the set of support vectors is the subset of informative examples that captures the necessary information for discrimination.

3.4.1.2 Mapping into Higher Dimensional Feature Spaces

For many problems, a linear partition of the feature space will not adequately approximate the true partition. Additional complexity in the hypothesis class may be required to achieve a more suitable approximation. Generally, more complex partitions are constructed implicitly in the original feature space by mapping the training examples into another space which simplifies the partitioning of the data. With neural network models, this is achieved by adding hidden units to a multi-layer perceptron or additional kernels to a radial basis function network. In this section, we examine a generalization of the linear machine that allows one to apply the same learning strategy in higher dimensional feature spaces.

Consider mapping vectors from the original feature space $\mathbf{X}$ to a higher dimensional feature space $\mathbf{Z}$ using the vector function $\Phi(X)$ prior to classification. Substituting the vector function into the equation for the classifier $g(X|\theta, \theta_b)$, we obtain the expression

$$g(X|\theta, \theta_b) = \operatorname{sgn} \left(\theta \cdot \Phi(X) + \theta_b \right) \tag{3.35}$$

for the generalized linear machine. In the last section, we learned that θ_*, the normal to the maximal margin hyperplane, can be expressed as a linear combination of the support vectors. Therefore the normal to the maximal margin hyperplane in the feature space $\mathbf{Z}$ can be expressed as

$$\theta_* = \sum_{i=1}^{N} \alpha_i Y^i \Phi \left(X^i \right). \tag{3.36}$$

Substituting this equation into equation 3.35, we obtain the expression

$$g(X|\theta_*, \theta_{b*}) = \operatorname{sgn} \left(\sum_{i=1}^{N} \alpha_i Y^i \left(\Phi \left(X^i \right) \cdot \Phi \left(X \right) \right) + \theta_{b*} \right) \tag{3.37}$$

for the *nonlinear support vector machine*. In order to determine the coefficients $\boldsymbol{\alpha}$ that define the maximal margin hyperplane in $\mathbf{Z}$, we can maximize the dual Lagrangian

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} Y^i Y^j \alpha_i \alpha_j \left(\Phi\left(X^i\right) \cdot \Phi\left(X^j\right) \right) \tag{3.38}$$

again subject to the constraints

$$\sum_{i=1}^{N} \alpha_i Y^i = 0, \ \alpha_i \geq 0 \quad i \in \{1, \ldots, N\}. \tag{3.39}$$

The support vectors in the feature space $\mathbf{Z}$ are simply the terms $\Phi\left(X^i\right)$ that correspond to the nonzero Lagrange multipliers α_i.

Notice that in both the learning and evaluation phases for the nonlinear SVM, the process can be defined solely in terms of dot products between points in the feature space $\mathbf{Z}$. There is no need to ever explicitly compute $\Phi(X)$. Therefore we can operate in high dimensional feature spaces defined by the transformation $\Phi(X)$ by defining the corresponding *kernel function*

$$K(U, V) = \Phi(U) \cdot \Phi(V). \tag{3.40}$$

Often algorithm designers specify the kernel function directly instead of working from a given mapping $\Phi(X)$. In order to ensure that the kernel function corresponds to an inner product, the kernel function must satisfy *Mercer's Condition* [11] which states that there exists a $\Phi(X)$ such that equation 3.40 holds if and only if the condition

$$\int K(U, V) f(U) f(V) dU dV \geq 0 \tag{3.41}$$

is satisfied for all $f(U)$ with finite L_2 norm. Example kernels that satisfy Mercer's condition are

$$K(U, V) = (U \cdot V + 1)^n \tag{3.42}$$

and

$$K(U, V) = \exp\left(\frac{-\|U - V\|^2}{\sigma^2} \right). \tag{3.43}$$

These kernel functions yield polynomial and radial basis function classifiers respectively. It is interesting to note that the dimensionality of the underlying feature space for the RBF classifier is infinite. Yet it is still possible to learn maximal margin hyperplanes that generalize well in such spaces from a finite training sample.

3.4.1.3 Soft Margin Optimization

In most real-world problems, we will not be able to partition the training examples without error due to the presence of noise in the data. Therefore some mechanism is needed to relax the margin constraints for examples that cannot be classified correctly. Typically this is accomplished by adding *margin slack variables* ξ_i [18] to the constraints and an associated penalty term to the objective function. With the addition of the margin slack variables, we obtain the *soft margin constraints*

$$Y^i \left(\theta \cdot \Phi\left(X^i\right) + \theta_b \right) \geq 1 - \xi_i \quad \xi_i \geq 0 \tag{3.44}$$

for the nonlinear support vector machine. ξ_i indicates the deviation from the minimum margin associated with the *ith* training example. Notice when $\xi_i > 1$, the example is classified incorrectly. Therefore $\sum_i \xi_i$ is an upper bound on the number of training errors.

With the addition of the penalty term to the objective function

$$\|\theta\|^2 + C\left(\sum_{i=1}^{N} \xi_i\right)^k, \tag{3.45}$$

the goal is no longer simply the maximization of the margin. Now there is a tradeoff between margin maximization and training error minimization that is controlled by the parameter C. Although the optimization problem is convex for $k \geq 1$, typically $k = 1$ (*1-norm soft margin*) or $k = 2$ (*2-norm soft margin*) since the optimization problem can be formulated as a quadratic program for these values. The optimal tradeoff parameter C is identified through cross-validation. In an alternative approach presented more recently by Schölkopf et al. [74], the tradeoff parameter represents a lower bound on the fraction of support vectors and an upper bound on the fraction of training errors.

3.4.1.4 Support Vector Learning for Multi-Class Problems

Up to this point, we have only discussed learning support vector machines for binary classification problems. Several approaches have been presented in the literature for addressing general $\mathcal{C}$ class classification problems. The most common approach involves simply training $\mathcal{C}$ binary support vector machines that discriminate between class k and the remaining classes [11]. Unfortunately, there is no bound on the generalization performance for this classifier [61]. Others [81, 83] have formulated the learning process as a single quadratic program and noted that this approach provides solutions to problems that cannot be solved by training $\mathcal{C}$ machines individually [2]. A more recent effort [61] has focused on the construction of a tree classifier composed of a collection of SVMs that discriminate between pairs of classes. Researchers continue to explore possible approaches to the multi-class problem and associated generalization bounds.

3.4.1.5 Suitability for Real-Time Image Sequence Classification

Support vector machines offer a powerful learning paradigm for problems that involve learning partitions in high dimensional spaces. The experiments conducted by Papageorgiou and Poggio [59, 60] provide a clear demonstration of this in the context of two object detection tasks. In [60], receiver operating characteristic (ROC) curves are presented for several second order polynomial SVMs trained on feature sets of various sizes derived from an overcomplete Haar wavelet dictionary. The interesting trend exhibited in these ROC curves is that the power of the detector *increases* as the complexity of the feature set increases. Using the full set of 1,326 wavelets to extract features from 1,848 positive and 11,361 negative training examples, the support vector approach yielded the most powerful detector. To emphasize further the capacity control achieved by SVMs, two other detectors were trained using the full set of wavelet features for training sets containing only one and ten positive examples. The ROC curves for these detectors are surprisingly comparable to the detectors trained using a reduced feature set and the entire set of positive examples. In [59], additional ROC curves are shown comparing the most powerful detector presented in [60] with another detector utilizing a high dimensional feature space with four times as

many dimensions. As before, we see improvements in performance obtained from increasing dimensionality.

The major disadvantage of nonlinear support vector machines is that they are computationally expensive to evaluate due to the fact that a large number of support vectors are generally used to represent the partition. Several approaches have been presented to ease this burden. Burges [8, 10] introduced a technique for approximating a support vector partition with a reduced set of vectors that are not training examples. This procedure involves specifying the size of the reduced set and minimizing the squared error between the actual and approximate normal to the hyperplane in feature space, which is a nontrivial optimization problem. More recently, Tipping [79] introduced a Bayesian formulation of the SVM that typically yields models requiring dramatically fewer kernel functions. The *relevance vectors* associated with the kernels also do not correspond to training examples. Unfortunately, any gains obtained through the sparsity of the model are offset by the Bayesian integration which is computationally expensive.

For applications such as the image sequence classification task with severe constraints on the computational resources, it will be difficult to learn a sparse generalized linear model of the partition that provides sufficient classification performance and computational efficiency. One of the appealing aspects of the support vector approach is that the formulation of the problem generally leads to a unique solution unlike backpropagation networks.[1] Unfortunately, uniqueness is lost once one attempts to learn a smaller set of arbitrary vectors that provides a more compact representation of the partition.

In order to satisfy the requirements for computational efficiency, especially in scenarios where the classifier must discriminate between a large number of classes, another classifier structure may be required. Decision trees [3, 61, 4] constructed from linear support vector machines may offer an appealing alternative. Given the fact that the number of support vectors does not affect the computational complexity of a linear SVM, a decision tree could provide significant computational advantages by reducing the average number of dot products required. We will investigate this structure in a later chapter.

3.4.2 Boosting

3.4.2.1 The Concept

In contrast to support vector learning, boosting developed from a fundamentally different line of research. Given a *weak* learning algorithm [49] that performs marginally better than random guessing, the question was raised by Kearns and Valiant [48] of whether it is possible to construct a *strong* learning algorithm [49] based upon a weak learning algorithm that is able to achieve an error rate arbitrarily close to the optimum with high confidence. In 1989, Schapire [70] introduced the first provably polynomial-time boosting algorithm. Since then, a variety of improved boosting algorithms have been introduced [28, 62, 55].

Boosting algorithms construct high performance classifiers by integrating a collection of *base classifiers* [71] learned using a weak learning algorithm. In the case of a binary classification problem, the base classifiers $h_t(X)$ map feature vectors from the space $\mathbf{X}$ to the set $\{-1, 1\}$. The *combined classifier* $H_M(X)$ then computes a weighted sum of the votes

[1] Burges and Crisp [9] have shown that when the SVM solution is not uniquely defined, the bias is the only parameter that must be identified through a line search.

from the M base classifiers of the form

$$H_M(X) = \text{sgn}\left(\frac{\sum_{t=1}^{M} \alpha_t h_t(X)}{\sum_{t=1}^{M} \alpha_t}\right) \quad (3.46)$$

$$= \text{sgn}\left(\sum_{t=1}^{M} \frac{\alpha_t}{\|\alpha\|_1} h_t(X)\right) \quad (3.47)$$

$$= \text{sgn}(\theta \cdot h(X)) \quad (3.48)$$

where $h(X) = [h_1(X)\ h_2(X)\ \ldots\ h_M(X)]^{\mathrm{T}}$, $\sum_t \theta_t = 1$ and $\theta_t \geq 0$. Notice that the combined classifier transforms a given feature vector X into a M-dimensional feature vector $h(X) \in \{-1, 1\}^M$ and partitions the M-dimensional feature space with a hyperplane.

In order to specify the combined classifier, we must define the set of base classifiers $h(X)$ and the weights θ. The most widely studied boosting algorithm, *AdaBoost* (*ADAptive BOOSTing*) [77], constructs the combined classifier in the following manner. AdaBoost specifies a distribution $D_t(i)$ that defines the influence of the training examples in the learning process during iteration t. Initially, this distribution is uniform. The distribution is provided to the weak learning algorithm if it can incorporate the distribution into the learning process. Otherwise, the training set is sampled with replacement according to the distribution, and the new training set is provided to the weak learning algorithm.

Once the weak learning algorithm returns the t^{th} base classifier $h_t(X)$, the corresponding weight θ_t is computed and the distribution is updated. θ_t is defined as

$$\theta_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) \quad (3.49)$$

where ϵ_t is the base classifier's error rate on the training set. The new distribution $D_{t+1}(i)$ is defined as

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\theta_t Y^i h_t(X^i))}{Z_t} \quad (3.50)$$

where Z_t is the normalization constant that insures D_{t+1} is a distribution. The exponential term increases the weight on a particular example when it is classified correctly and decreases the weight when it is classified incorrectly. The rate of change in the weights is determined by the training error of the base classifier. Through this process, AdaBoost focuses the learning on the training examples that are most difficult for the base classifiers to classify correctly.

As noted by Schapire et al. [71] and others, AdaBoost exhibits resistance to overfitting on a variety of problems as the number of base classifiers is increased. In the example discussed in [71], the training error reached zero after five iterations. Yet the test error continued to decrease even after 100 iterations. The combined classifier resulting after 1000 iterations outperformed the classifier obtained after five iterations. Given the complexity of the combined classifier containing 1000 base classifiers, this result is rather surprising. Yet as we have observed with support vector machines, the number of parameters does not always provide an effective measure of complexity.

The effectiveness of AdaBoost can be better understood when one considers the influence of AdaBoost on the margin distribution as the number of iterations increases. Freund and

Schapire [28] have shown that AdaBoost reduces the fraction of training examples with margins less than a small positive constant exponentially fast as the number of iterations increases. By expressing the distribution values $D_{T+1}(i)$ in terms of the original distribution values $D_1(i)$, we also discover that

$$D_{T+1}(i) = \frac{D_1(i)\exp(-\theta_1 Y^i h_1(X^i))\cdots\exp(-\theta_T Y^i h_T(X^i))}{Z_1\cdots Z_T} \quad (3.51)$$

$$= \frac{D_1(i)\exp\left(-Y^i\sum_{t=1}^{T}\theta_t h_t(X^i)\right)}{Z_1\cdots Z_T} \quad (3.52)$$

which indicates AdaBoost places the most weight on the training examples that produce the smallest margins

$$\gamma_T(i) = Y^i\sum_{t=1}^{T}\theta_t h_t(X^i) \quad (3.53)$$

from the combined classifier during iteration T. Therefore the success of AdaBoost can be attributed to the capacity control achieved by incrementally *increasing* the dimensionality of the feature space with the intent of maximizing the margins.

Recently, an alternative view of boosting has been presented by Mason et al. [55] which illustrates that a large number of boosting algorithms perform gradient descent in function space with respect to some risk function

$$R(H_T) = \frac{1}{N}\sum_{i=1}^{N} C(\gamma_T(i)) \quad (3.54)$$

of the margin. In the case of AdaBoost, the specific risk function minimized is

$$R_{AB}(H_T) = \frac{1}{N}\sum_{i=1}^{N} e^{-\gamma_T(i)}. \quad (3.55)$$

Due to AdaBoost's emphasis on the training examples with the smallest margins, AdaBoost has difficulty with noisy training sets. Mason et al. [55, 54] have introduced several parameterized families of monotonically decreasing cost functions $C(\gamma,\lambda)$ of the margin that approximate the error counting function

$$\frac{1}{2N}\sum_{i=1}^{N} 1 - \mathrm{sgn}(\gamma_T(i)) \quad (3.56)$$

and offer improvements over AdaBoost. For families of cost functions such as those introduced in [55, 54], a generalization bound has also been developed that expresses the tradeoff between minimizing the training error and maximizing the margins as a function of λ.

DOOM II [55] is one of the latest boosting algorithms that provides improved robustness to noise over AdaBoost. DOOM II employs a sigmoidal approximation of the error counting function of the form

$$R_{D2}(H_T) = \frac{1}{N}\sum_{i=1}^{N}(1-\tanh(\lambda\gamma_T(i)) \quad (3.57)$$

$$= \frac{1}{N}\sum_{i=1}^{N}\frac{2}{1+e^{2\lambda\gamma_T(i)}} \quad (3.58)$$

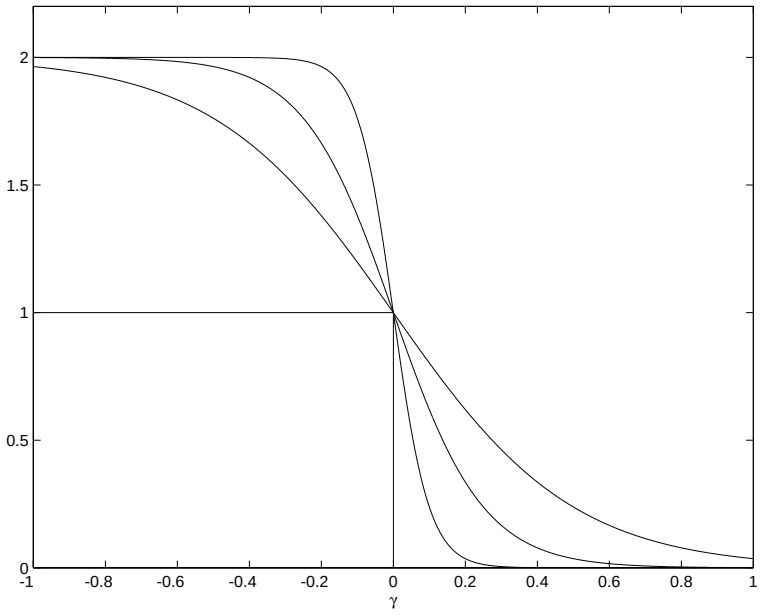


Figure 3.2: The error counting function and DOOM II cost function for $\lambda = 2, 4, 10$

as shown in figure 3.2 for several values of λ. As $\lambda \to \infty$, minimization of $R_{D2}(H_T)$ corresponds to empirical risk minimization. For finite values of λ, minimizing the risk function places some amount of emphasis on margin maximization to control the capacity. Since the slope of the cost functions shown in figure 3.2 tends toward zero for large margins, DOOM II provides an advantage over AdaBoost in that examples with large negative margins do not dominate the learning procedure. The range of margins over which training examples have significant influence is controlled directly by the parameter λ.

3.4.2.2 Suitability for Real-Time Image Sequence Classification

Although the latest boosting algorithms have demonstrated excellent performance on a number of classification problems, the resulting combined classifiers often have a large number of base classifiers. Therefore the computational cost of evaluating the combined classifier is high. This result is not surprising given the nature of the learning procedure. By training the base classifiers using a weak learning algorithm and selecting the base classifier weights based on the training error rate, the available complexity in the combined classifier is most likely not being utilized efficiently to maximize the margins of the training examples. In order to learn a robust, computationally efficient classifier, we need to explicitly adapt the parameters of the classifier with respect to the risk function of the margin. In this way, we ensure that the available complexity in the model is being used efficiently to achieve the intended objective. We consider such a strategy in the next section.

3.4.3 Differential Learning

In order to design classifiers that generalize well, it is imperative that one does not attempt to solve a more general problem along the path toward a particular solution. This is one of the core philosophical principles that forms the foundation of differential learning and support vector learning [44][82, Ch. 9]. Earlier in this chapter, we questioned the value of learning probability distributions in image space as a means to approximate the

Bayes-optimal partition. We will now consider the limitation of probabilistic learning for classification and a general approach for learning approximations to the Bayes-optimal partition directly.

3.4.3.1 Approximating the Bayes-Optimal Partition

The Bayes-optimal partition of a feature space associates a given feature vector X with the class label ω_* producing the largest *a posteriori* probability $\mathrm{P}(\omega_*|X)$. As noted in section 3.2.1, a partition can be expressed in terms of a set of $\mathcal{C}$ indicator functions $\mathcal{I}_{\omega_k}(X)$ which indicate whether a given example X is within one of the $\mathcal{C}$ decision regions R_k. The indicator functions $\mathcal{I}^{Bayes}_{\omega_k}(X)$ for the Bayes-optimal partition can be expressed as

$$\mathcal{I}^{Bayes}_{\omega_k}(X) = \begin{cases} 1 & \text{if } \mathrm{P}(\omega_k|X) - \max\limits_{i,i\neq k} \mathrm{P}(\omega_i|X) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.59}$$

In this form of the indicator functions, the set of discriminant functions used to parameterize the partition are the *a posteriori* probabilities $\mathrm{P}(\omega_k|X)$. The probabilistic approach to approximating the Bayes-optimal partition involves estimating the *a posteriori* probabilities throughout feature space from the training examples and using those estimates $\hat{\mathrm{P}}(\omega_k|X)$ to define the $\mathcal{C}$ indicator functions.

The weakness of the probabilistic approach stems from the fact that the Bayes-optimal partition is approximated indirectly by addressing the more difficult problem of estimating the *a posteriori* probabilities. Accurate probability estimates are not required to design a classifier that generalizes well. Therefore by insisting that the discriminant functions correspond to estimates of the *a posteriori* probabilities, we are placing a significant constraint on the admissible discriminant functions which can be detrimental to our objective.

In order for a set of discriminant functions to yield the Bayes-optimal partition, the only constraint that must be satisfied is that the discriminant function producing the largest output always corresponds to the class label with the largest *a posteriori* probability. This implies that there are an infinite number of ways to parameterize the Bayes-optimal partition. To see this, consider transforming the *a posteriori* probabilities $\mathrm{P}(\omega_k|X)$ using a strictly monotonically increasing function $f(\beta)$ to generate a new set of discriminant functions

$$h_k(X) = f\left(\mathrm{P}(\omega_k|X)\right). \tag{3.60}$$

Clearly when

$$\mathrm{P}(\omega_k|X) - \max_{i,i\neq k} \mathrm{P}(\omega_i|X) > 0, \tag{3.61}$$

the inequality

$$f(\mathrm{P}(\omega_k|X)) - f\left(\max_{i,i\neq k} \mathrm{P}(\omega_i|X)\right) > 0 \tag{3.62}$$

$$h_k(X) - \max_{i,i\neq k} h_i(X) > 0 \tag{3.63}$$

also holds. Therefore these parameterizations yield the same underlying partition. Since there are an infinite number of strictly monotonically increasing functions, the set of all possible sets of discriminant functions yielding Bayes-optimal classification is infinite as well.

By removing the probabilistic constraint on the discriminant functions, we effectively increase the capacity of a given hypothesis class. This has two implications. The hypothesis

class now admits sets of discriminant functions that can properly partition the feature space with reduced functional complexity as compared to the probabilistic discriminant functions [44]. Yet increasing the capacity may also negatively impact generalization performance. Therefore we will have to control the capacity of the hypothesis class in other ways.

To approximate the Bayes-optimal partition directly, the strategy will be to learn a set of discriminant functions that maximize the *discriminant differentials* of the training examples. The discriminant differential $\delta(X|\theta)$ is defined as the difference between the discriminant function associated with the correct class and the largest other discriminant function. For a training image X with class label ω_k, the discriminant differential is denoted as

$$\delta_k(X|\theta) = g_k(X|\theta) - \max_{i,i\neq k} g_i(X|\theta). \tag{3.64}$$

Notice that when X is correctly classified, $g_k(X|\theta)$ produces the largest output and $\delta_k(X|\theta)$ is positive. The concept of the discriminant differential can be thought of as a generalization of the margin for general $\mathcal{C}$ class classification problems. Capacity control will be achieved by maximizing the discriminant differentials of the training examples using a differentiable objective function similar to those introduced by Mason et al. [55, 54]. In the next section, we will define the conditions that the objective function must satisfy to induce large differential partitions and admit the Bayes-optimal partition in the limit.

3.4.3.2 The Classification Figure-of-Merit Objective Function

The class of objective functions we wish to consider are of the form

$$\mathrm{CFM}\left(\mathcal{S}^N|\theta\right) = \frac{1}{N}\sum_{i=1}^{N}\sigma(\delta_k(X^i|\theta),\psi) \tag{3.65}$$

where $\sigma(\delta,\psi)$ is a monotonically increasing function of the discriminant differential δ and the class label for the i^{th} example $\omega^i = \omega_k$. The parameter ψ indexes the objective functions within the defined class. We will first examine the outcome of maximizing the objective function in the limit of infinite training data. Our presentation is a variation on the original formulation presented by Hampshire [44].

As the number of training examples N becomes asymptotically large, the sample average $\mathrm{CFM}\left(\mathcal{S}^N|\theta\right)$ converges in probability to the expected value of CFM over $\mathbf{X}\times\Omega$ where Ω is the class label space. Expanding $\mathrm{E}_{\mathbf{X},\Omega}[\sigma(\delta(X|\theta),\psi)]$, we obtain

$$\begin{aligned}\mathrm{E}_{\mathbf{X},\Omega}[\sigma(\delta(X|\theta),\psi)] &= \int \mathrm{E}_{\Omega|\mathbf{X}}[\sigma(\delta(X|\theta),\psi)]\rho(\mathrm{X})\mathrm{dX} &(3.66)\\ &= \int \underbrace{\sum_{c=1}^{\mathcal{C}}\mathrm{P}(\omega_c|X)\sigma(\delta_c(X|\theta),\psi)}_{\mathrm{CFM}(X|\theta)}\rho(X)\mathrm{dX}. &(3.67)\end{aligned}$$

In order to maximize the expected value for all possible distributions over the feature space, we must maximize $\mathrm{CFM}(X|\theta)$ for all $X\in\mathbf{X}$. Given the coupled nature of the differentials δ_c, we accomplish this by first reexpressing $\mathrm{CFM}(X|\theta)$ in terms of the ranked differentials $\delta_{(k)}(X|\theta)$ where

$$\delta_{(1)} \geq \delta_{(2)} \geq \ldots \geq \delta_{(\mathcal{C})}. \tag{3.68}$$

Since

$$\delta_{(1)} = g_{(1)}(X|\theta) - g_{(2)}(X|\theta), \tag{3.69}$$

$$\delta_{(k)} = g_{(k)}(X|\theta) - g_{(1)}(X|\theta), \tag{3.70}$$

we can express $\delta_{(k)}$ for $k > 1$ as

$$\delta_{(k)} = -\delta_{(1)} - \epsilon_{(k)} \tag{3.71}$$

where

$$\epsilon_{(k)} = g_{(2)}(X|\theta) - g_{(k)}(X|\theta). \tag{3.72}$$

This implies $\mathrm{CFM}(X|\theta)$ equals

$$\mathrm{CFM}(X|\theta) = \sum_{i=1}^{\mathcal{C}} \mathrm{P}(\omega_{(i)}|X)\sigma(\delta_{(i)}(X|\theta), \psi) \tag{3.73}$$

$$\begin{aligned} = \quad & \mathrm{P}(\omega_{(1)}|X)\sigma(\delta_{(1)}(X|\theta), \psi) + \\ & \sum_{i=2}^{\mathcal{C}} \mathrm{P}(\omega_{(i)}|X)\sigma(-\delta_{(1)}(X|\theta) - \epsilon_{(i)}(X|\theta), \psi). \end{aligned} \tag{3.74}$$

Since $\frac{\partial \sigma}{\partial \delta} \geq 0$ and $\epsilon_{(k)} \geq 0$, equation 3.74 is maximized when all discriminant function outputs other than the maximum are equal. In this scenario, $\epsilon_{(k)} = 0$ and $\mathrm{CFM}(X|\theta)$ becomes

$$\mathrm{CFM}(X|\theta) = \mathrm{P}(\omega_{(1)}|X)\sigma(\delta_{(1)}(X|\theta), \psi) + (1 - \mathrm{P}(\omega_{(1)}|X))\sigma(-\delta_{(1)}(X|\theta), \psi). \tag{3.75}$$

Given that

$$\sigma(\delta_{(1)}(X|\theta), \psi) \geq \sigma(-\delta_{(1)}(X|\theta), \psi), \tag{3.76}$$

equation 3.75 is maximum when the largest discriminant function output corresponds to the most likely class ω_* which implies

$$\mathrm{CFM}(X|\theta) = \mathrm{P}(\omega_*|X)\sigma(\delta_*(X|\theta), \psi) + (1 - \mathrm{P}(\omega_*|X))\sigma(-\delta_*(X|\theta), \psi). \tag{3.77}$$

In order to induce the Bayes-optimal partition for any distribution $\rho(X)$ given sufficient complexity in the hypothesis class, $\sigma(\delta, \psi)$ must be of the form

$$\sigma(\delta, \psi) = \begin{cases} \alpha & \text{if } \delta > 0 \\ \beta & \text{otherwise} \end{cases} \tag{3.78}$$

where $\alpha > \beta$. Substituting the step form of $\sigma(\delta, \psi)$ into the expectation in equation 3.67, we find

$$\mathrm{E}_{\mathbf{X},\Omega}[\sigma(\delta(X|\theta), \psi)] = \int \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_c|X)\sigma(\delta_c(X|\theta), \psi)\rho(X)\mathrm{dX} \tag{3.79}$$

$$= \int \sum_{i=1}^{\mathcal{C}} \mathrm{P}(\omega_{(i)}|X)\sigma(\delta_{(i)}(X|\theta), \psi)\rho(X)\mathrm{dX} \tag{3.80}$$

$$= \int \left[\alpha \mathrm{P}(\omega_{(1)}|X) + \beta(1 - \mathrm{P}(\omega_{(1)}|X))\right] \rho(X)\mathrm{dX} \tag{3.81}$$

$$= \int \left[\beta + (\alpha - \beta)\mathrm{P}(\omega_{(1)}|X)\right] \rho(X)\mathrm{dX} \tag{3.82}$$

$$= \beta + (\alpha - \beta)\mathrm{P}(\omega_{(1)}) \tag{3.83}$$

which indicates that the expected value of CFM is a monotonically increasing, linear function of the probability of correct classification $\mathrm{P}(\omega_{(1)})$ for the partition specified by the parameter vector θ. Therefore maximizing the expectation with respect to θ such that $\omega_{(1)} = \omega_*$ yields the Bayes-optimal partition.

Notice that learning with a step form of CFM corresponds to empirical risk minimization. Empirical risk minimization is appropriate in the setting we have described where the training set becomes asymptotically large. Yet it should be avoided when only limited training data is available unless some mechanism for capacity control is employed. We will consider other forms of CFM that are approximations of the step form and induce large discriminant differentials to control the capacity of the hypothesis class. We begin by deriving conditions that must be satisfied in order for CFM to favor partitions which induce large discriminant differentials.

Let us return to equation 3.77 obtained from the maximization of $\mathrm{CFM}(X|\theta)$. If we want the objective function to favor partitions with large discriminant differentials, we would like $\mathrm{CFM}(X|\theta)$ to be monotonically increasing with respect to the differential δ_*. Therefore we will insist that

$$\begin{aligned}\mathrm{P}(\omega_*|X)\sigma(\delta_a,\psi) + (1-\mathrm{P}(\omega_*|X))\sigma(-\delta_a,\psi) \geq \\ \mathrm{P}(\omega_*|X)\sigma(\delta_b,\psi) + (1-\mathrm{P}(\omega_*|X))\sigma(-\delta_b,\psi)\end{aligned} \tag{3.84}$$

when $\delta_a > \delta_b > 0$. Manipulating both sides of the inequality, we obtain the condition

$$\frac{\sigma(\delta_a,\psi) - \sigma(\delta_b,\psi)}{\sigma(-\delta_b,\psi) - \sigma(-\delta_a,\psi)} \geq \frac{1-\mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)}. \tag{3.85}$$

Comparing this inequality with the condition

$$\frac{\sigma(\delta,\psi) - \sigma(0,\psi)}{\sigma(0,\psi) - \sigma(-\delta,\psi)} \geq \frac{1-\mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)} \tag{3.86}$$

obtained by Hampshire [44, Ch. 2] in his original investigation of CFM, we see that the imposition of the monotonicity constraint places stronger constraints on $\sigma(\delta,\psi)$ over the entire range of the differential δ. Hampshire obtained inequality 3.86 by insisting only that $\mathrm{CFM}(X|\theta) > \sigma(0,\psi)$ when $\delta_* > 0$. Therefore it is possible that some partitions may induce a lower CFM than other partitions yielding smaller differentials when only inequality 3.86 is enforced.

We now derive conditions on the partial derivative of $\sigma(\delta,\psi)$ with respect to δ from inequality 3.85. If we set $\delta_a = \delta_b + \delta$, divide the numerator and denominator of the left hand side by δ and take the limit as $\delta \to 0^+$, we obtain

$$\lim_{\delta\to 0^+} \frac{\frac{\sigma(\delta_b+\delta,\psi)-\sigma(\delta_b,\psi)}{\delta}}{\frac{\sigma(-\delta_b,\psi)-\sigma(-\delta_b-\delta,\psi)}{\delta}} \geq \frac{1-\mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)} \tag{3.87}$$

$$\frac{\frac{\partial}{\partial\delta}\sigma(\delta_b,\psi)}{\frac{\partial}{\partial\delta}\sigma(-\delta_b,\psi)} \geq \frac{1-\mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)}. \tag{3.88}$$

This general constraint on the ratio of the derivatives at $\delta = \delta_b$ and $\delta = -\delta_b$ for all δ_b includes the constraint

$$\frac{\frac{\partial}{\partial\delta}\sigma(0^+,\psi)}{\frac{\partial}{\partial\delta}\sigma(0^-,\psi)} \geq \frac{1-\mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)} \tag{3.89}$$

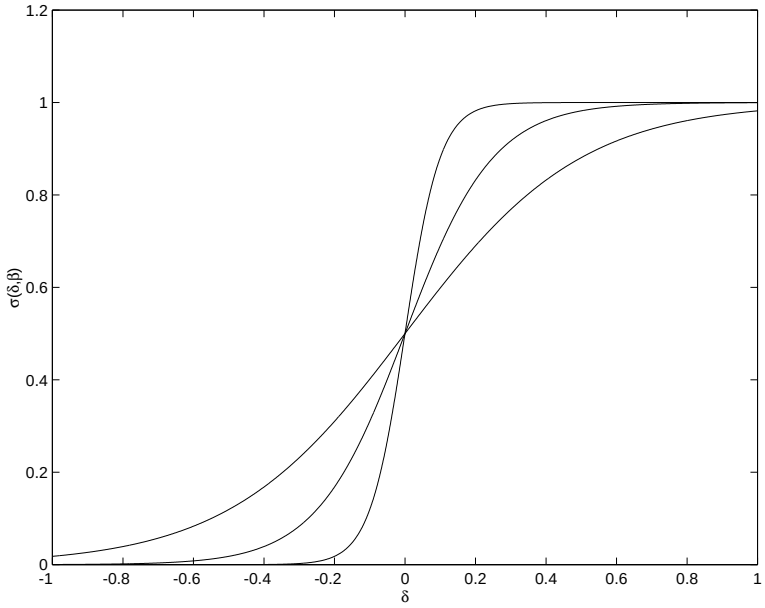


Figure 3.3: The sigmoidal form of CFM for $\alpha = 1, \zeta = 0$ and $\beta = 4, 8, 20$

which can be derived from Hampshire's inequality. Since $\mathrm{P}(\omega_*|X)$ lies within the interval $[\frac{1}{\mathcal{C}}, 1]$ where $\mathcal{C}$ is the number of classes, the right hand side of inequality 3.88 is bounded by

$$\frac{1 - \frac{1}{\mathcal{C}}}{\frac{1}{\mathcal{C}}} \geq \frac{1 - \mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)} \tag{3.90}$$

$$\mathcal{C} - 1 \geq \frac{1 - \mathrm{P}(\omega_*|X)}{\mathrm{P}(\omega_*|X)}. \tag{3.91}$$

Therefore in order for the monotonicity constraint to be satisfied for all possible distributions, the inequality

$$\frac{\frac{\partial}{\partial \delta}\, \sigma(\delta_b, \psi)}{\frac{\partial}{\partial \delta}\, \sigma(-\delta_b, \psi)} \geq \mathcal{C} - 1 \tag{3.92}$$

must be satisfied for all $\delta_b \in (0, 1]$.

Given these constraints on the derivatives of CFM, we now consider two classes of CFM objective functions introduced by Hampshire. The original sigmoidal form of CFM studied by Hampshire in [43, 45] is defined as

$$\sigma(\delta, \alpha, \beta, \zeta) = \frac{\alpha}{1 + e^{-\beta\delta + \zeta}}. \tag{3.93}$$

Notice that this class of objective functions admits the DOOM II objective function introduced in equation 3.58. The sign difference between the expressions is due to the fact that DOOM II minimizes the cost function and differential learning maximizes CFM.

As in [43, 45], we will restrict ourselves to the subclass of objective functions where $\alpha = 1$ and $\zeta = 0$. Without loss of generality, the discriminant function outputs are assumed to lie in the interval [0,1] such that the discriminant differential $\delta \in [-1, 1]$. The sigmoidal form of CFM is shown in figure 3.3 for several values of β.

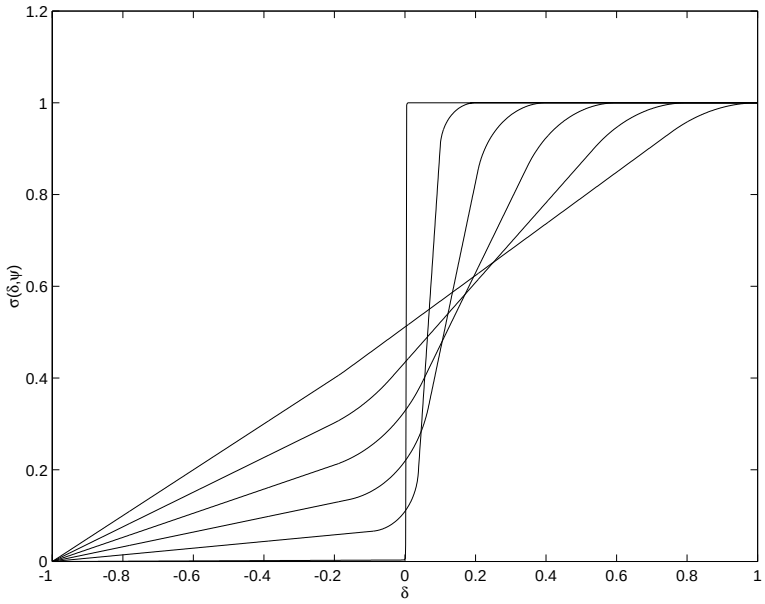


Figure 3.4: The synthetic form of CFM for several values of confidence ψ

One can show that for the sigmoidal form of CFM,

$$\frac{\frac{\partial}{\partial\delta}\,\sigma(\delta_b,\beta)}{\frac{\partial}{\partial\delta}\,\sigma(-\delta_b,\beta)} = 1 \tag{3.94}$$

for all admissible values of δ_b and β. This implies that condition 3.92 is only satisfied when $\mathcal{C} = 2$. Therefore the sigmoidal class of objective functions yields monotonicity independent of the underlying distribution only for two class problems.[2] For general $\mathcal{C}$ class problems, another class of objective functions is needed.

In order to obtain a more general family of objective functions, Hampshire [44] designed a synthetic form of CFM which is defined piecewise in terms of line segments and circular arcs and parameterized in terms of the *confidence parameter* ψ. By increasing ψ over the interval [0,1], the synthetic form of CFM varies between a step function and a line as depicted in figure 3.4. For $\delta \geq \psi$, the synthetic form of CFM equals one.

As figure 3.4 highlights, the synthetic form offers increased flexibility in that the ratio of the derivatives can be made arbitrarily large as necessitated by inequality 3.92 for a majority of the differential interval $[0, \psi]$ as ψ is reduced. Unfortunately, reduction of ψ also has the negative effect of increasing the differential interval $[\psi, 1]$ over which the inequality 3.88 is likely violated. Since the ratio of the derivatives for $\delta \geq \psi$ equals zero, the only way the monotonicity constraint can be satisfied is if $\mathrm{P}(\omega_*|X) = 1$. Therefore confidence reduction must be pursued with care. By varying the confidence parameter, we are adjusting the tradeoff between capacity control and training error minimization. Cross-validation provides the practical tool for selecting an appropriate compromise between the two objectives.

Although the synthetic form of CFM has been shown to yield classifiers that offer excellent performance on a variety of problems and improved rates of learning over the sigmoidal form [44], there are weaknesses associated with the synthetic form that need to be studied

[2] It is interesting to note that DOOM II is evaluated only on two class problems in [55].

further. In problems where there is significant overlap of the class-conditional densities, Hampshire[3] has observed experimentally that antisymmetric forms of CFM similar to the sigmoidal form provide superior performance over the synthetic form in various two class problems. It will be interesting to explore whether the success of antisymmetric forms can be attributed to the fact that they satisfy inequality 3.92 over the entire range of differentials in contrast to the synthetic form.

3.4.3.3 Suitability for Real-Time Image Sequence Classification

Differential learning provides several advantages over support vector learning and boosting that make the learning procedure more attractive for our application. The most significant advantage comes from the ability to control the complexity of the classifier architecture. In contrast to support vector learning and boosting where the classifier architecture is determined to some degree by the learning procedure, differential learning requires the user to specify the classifier architecture prior to learning. In general, this may be viewed as a disadvantage. Yet it allows us to explicitly search for large differential partitions within hypothesis classes with low computational complexity. At the same time, differential learning offers improved robustness to noise over support vector learning and a general approach to large margin classification for multi-class problems. As we will see in the following chapter, differential learning also allows one to adapt parameterized representations with respect to the objective function. In this way, one can jointly optimize the representation and partition to achieve large differentials in the feature space. Given these distinct advantages, we will use differential learning to learn large differential partitions in image and class label distribution space.

3.5 Managing the Adverse Effects of Dependent Data and Unknown Class Prior Probabilities

All of the learning procedures we have discussed in this chapter assume that the process giving rise to the training sample is stationary and the training sample results from a series of independent, identically distributed trials. In the case of the image and class label distribution partitioning problems, these assumptions are violated to varying degrees. The collection of images used to train the image classifier does not result from a series of independent trials. In addition, the class prior probabilities often vary significantly over time and across environments. Therefore we must consider what modifications to the learning procedure are needed to reduce the likelihood of learning partitions that fail to offer adequate performance across a broad range of environments.

3.5.1 Learning from Dependent Image Data

Generally when learning a classifier, the objective is to minimize the classifier's error rate. Our ultimate objective is to design a pair of classifiers that collectively yields a low image sequence error rate; thus, one may question whether the image error rate is the ideal performance measure for the image classifier due to the dependencies in the image data. Ideally, we would like to minimize the number of image misclassifications in a given sequence. Therefore an alternate performance measure to consider is the *sequence image error rate* $P_{e_{SI}}$ which is the average fraction of images classified incorrectly in a sequence. The sample

[3] Private communication with John Hampshire.

average sequence image error rate $\hat{\mathrm{P}}_{\mathrm{e_{SI}}}$ is expressed as

$$\hat{\mathrm{P}}_{\mathrm{e_{SI}}} = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{1}{N_{S_i}} \sum_{j=1}^{N_{S_i}} 1 - \mathcal{I}_{\omega_k}(S_i{}^j|\theta) \tag{3.95}$$

where N_S is the number of sequences, N_{S_i} is the number of images in sequence S_i and ω_k is the class label for the i^{th} image sequence. This measure is intuitively appealing in that it offers a more meaningful assessment of performance relative to our overall objective.

Since the synthetic CFM objective function can be viewed as a differentiable approximation to the correct classification counting function, minimizing the sequence image error rate within the framework of differential learning can be achieved by simply modifying the averaging process over the training sample. When assuming the sequence images are independent, the sample average CFM is expressed as

$$\mathrm{CFM}\left(\mathcal{S}^N|\theta\right) = \frac{1}{N} \sum_{i=1}^{N_S} \sum_{j=1}^{N_{S_i}} \sigma\left(\delta_k\left(S_i{}^j|\theta\right), \psi\right) \tag{3.96}$$

where ω_k is the class label for the i^{th} image sequence. In order to reweight the contributions from each sequence, we will perform a two-step averaging process as in equation 3.95. First we will average the contributions over each sequence. Then we will average across sequences. This yields the objective function

$$\mathrm{CFM}_{Seq}\left(\mathcal{S}^N|\theta\right) = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{1}{N_{S_i}} \sum_{j=1}^{N_{S_i}} \sigma\left(\delta_k\left(S_i{}^j|\theta\right), \psi\right). \tag{3.97}$$

For each of these objective functions, we can imagine scenarios where one objective function will provide clearly superior performance. Therefore it is not obvious that one objective function will provide a distinct advantage over the other across a range of possible distributions. Given the lengths of the image sequences are dependent on the nature of the data collection, we are inclined to average over the images as in equation 3.96 to avoid imposing additional biases during learning. If the experimental results prove to be disappointing, we will investigate minimizing the sequence image error rate using equation 3.97 as well.

Although we will not attempt to directly minimize the sequence image error rate over the training sample, we will evaluate a bound on the sequence image error rate of the classifier on a validation set during learning to avoid overfitting. In the next section, we introduce the procedure used to bound the performance of the classifier when the class prior probabilities are unknown.

3.5.2 Minimizing the Worst Case Performance

When optimizing error measures or objective functions of the margin/differential, the influence of each class during learning is partially determined by the number of examples of each class present in the training data. Assuming that the training sample results from a series of independent, identically distributed trials, we expect that the fraction of examples in the training sample corresponding to a given class provides an estimate of the true class prior probability. Therefore when the true class prior probabilities are unknown and variable, we need to consider how to properly bias the learning so that we avoid poor performance in malicious environments.

In appendix B, we examine the general problem of minimizing the maximum class-conditional error rate and derive a condition that must be satisfied in order to obtain a *minimax partition*. A minimax partition is simply a partition with class-conditional error rates that are all equal. Then we investigate a variation on differential learning entitled *minimax differential learning* which induces a minimax partition in the limit of infinite training data as the confidence parameter ψ is annealed to zero. The fundamental difference between differential learning and minimax differential learning is that minimax differential learning maximizes the minimum class-conditional CFM instead of the sample average CFM. We will initially employ differential learning to learn the image and class label distribution partitions. If the results leave significant margins for improvement, we will explore the potential benefits of using minimax differential learning to learn minimax partitions directly.

During learning, we will evaluate classifier performance by computing an upper bound on the *worst case error rate* on the validation set. Worst case classification performance is obtained in an environment where the classifier is provided only with examples from the class yielding the maximum class-conditional error rate. In the ideal case where a large amount of validation data is available from each class, a reliable estimate of the worst case error rate could be obtained by simply computing the maximum of the class-conditional error rates on the validation set. In the more common scenario where we have a limited amount of validation data, the maximum class-conditional error rate may be a poor estimate of the true maximum error rate due to the small sample size. Therefore we compute upper bounds with 95% confidence on the class-conditional error rates and select the maximum upper bound as our performance measure so that the uncertainty in the class-conditional error rate estimates is not discounted. *Hoeffding's inequality* is employed to obtain the upper bound. See appendix A for a derivation of the confidence bound.

3.6 Confidence Assessment and Rejection

The difficulties that arise when the training sample is not representative of the underlying data distribution complicate the task of classification confidence assessment as well. In this section, we investigate the problem with the standard approach of estimating the probability of correct classification. Since the rejection region is typically defined based on the probability of correct classification, we then propose an alternative procedure for defining the rejection region. Finally, we address the problem of defining a measure of classification confidence to rank the image sequences.

3.6.1 Assessing Classification Confidence

In order to evaluate confidence in a classification decision, a reliable estimate of the probability of correct classification $\mathrm{P}(\omega_*|X)$ is required. Estimating the probability of correct classification necessitates the estimation of the *a posteriori* probabilities throughout feature space. As we have discussed earlier, probabilistic learning is not a viable option when assuming independence. Even if probabilistic learning was appropriate, we question the effectiveness of the probability of correct classification as a confidence measure when our model of the underlying data distribution deviates significantly from the true distribution.

To motivate our concern, we examine the following three class problem. X is a measurement of a constant signal corrupted by zero mean, additive, white Gaussian noise. The underlying constant signal can take on one of three values. Only two of these values are known. The variance of the noise is also known. Due to the nature of our measurement procedure, we know beforehand that outliers may appear in our measurements. Therefore we want to design a classifier that can reliably reject examples that deviate significantly

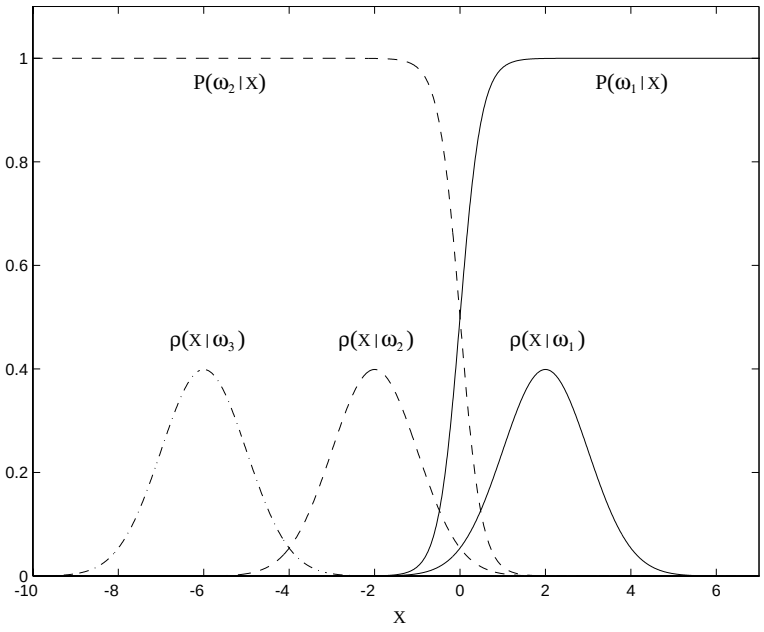


Figure 3.5: The class-conditional densities and *a posteriori* probabilities for the hypothetical three class problem

from the known class-conditional densities $\rho(X|\omega_1)$ and $\rho(X|\omega_2)$ shown in figure 3.5. We will assume that the known classes are equally likely prior to making the measurement.

If we design the Bayes-optimal classifier based on the assumption of two possible classes ω_1 and ω_2, we find the *a posteriori* probabilities $\mathrm{P}(\omega_k|X)$ equal

$$\mathrm{P}(\omega_k|X) = \frac{\rho(X|\omega_k)\mathrm{P}(\omega_k)}{\rho(X|\omega_1)\mathrm{P}(\omega_1) + \rho(X|\omega_2)\mathrm{P}(\omega_2)} \tag{3.98}$$

$$= \frac{\rho(X|\omega_k)}{\rho(X|\omega_1) + \rho(X|\omega_2)}. \tag{3.99}$$

This implies the Bayes-optimal class label ω_* equals

$$\omega_* = \underset{\omega\in\{\omega_1,\omega_2\}}{\operatorname{argmax}} \mathrm{P}(\omega|X) = \underset{\omega\in\{\omega_1,\omega_2\}}{\operatorname{argmax}} \rho(X|\omega) \tag{3.100}$$

and the probability of correct classification $\mathrm{P}(\omega_*|X)$ equals

$$\mathrm{P}(\omega_*|X) = \frac{\rho(X|\omega_*)}{\rho(X|\omega_1) + \rho(X|\omega_2)}. \tag{3.101}$$

Figure 3.5 shows the *a posteriori* probabilities $\mathrm{P}(\omega_k|X)$ as a function of the measurement value.

Notice that the probability of correct classification $\mathrm{P}(\omega_*|X)$ decreases to the minimum confidence value only about $X = 0$. This is due to the fact that the normalization term in the denominator of equation 3.98 is incorrect. The normalization term is the unconditional density $\rho(X)$. Since we do not have complete knowledge of the classes, the sum

$$\hat{\rho}(X) = \mathrm{P}(\omega_1)\rho(X|\omega_1) + \mathrm{P}(\omega_2)\rho(X|\omega_2) \tag{3.102}$$

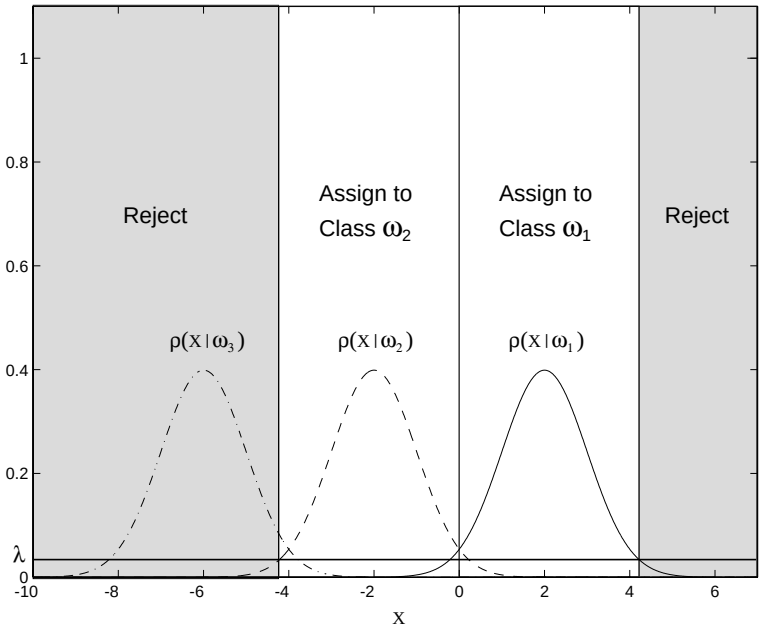


Figure 3.6: Decision regions induced by thresholding the known class-conditional densities $\rho(X|\omega_k)$

represents an incomplete model of $\rho(X)$. Therefore the *a posteriori* probabilities are unreliable in regions of the measurement space where the approximate unconditional density $\hat{\rho}(X)$ deviates significantly from the true unconditional density $\rho(X)$. This indicates that in domains such as surveillance where the training sample is not representative of the underlying distribution $\rho(X)$, estimating the probability of correct classification will not provide a meaningful estimate of classification confidence.

3.6.2 Defining the Rejection Region

When we are unable to estimate the probability of correct classification, we must devise another strategy for defining the rejection region. Our objective will be to constrain the decision regions to regions of feature space where the training data is contained. By minimizing the size of the decision regions while limiting the fraction of correctly classified examples that are rejected, we are increasing the likelihood of rejecting ambiguous and unknown examples without a significant impact on performance. Within the context of the toy problem, this objective is easily achieved by thresholding the *maximum likelihood*

$$\rho(X|\omega_*) = \max_{\omega \in \{\omega_1, \omega_2\}} \rho(X|\omega). \tag{3.103}$$

As illustrated in figure 3.6, thresholding $\rho(X|\omega_*)$ induces closed decision regions about the modes of the class-conditional densities and minimizes erroneous classifications in portions of measurement space with low likelihoods.

In the context of the image space partitioning problem, we want to avoid estimating the class-conditional densities in order to constrain the decision regions. Instead we would like to learn the decision regions directly from the data. This objective is related to the task of estimating the support of a distribution which involves estimating an indicator function that is positive in the region containing the majority of the data and negative elsewhere. This

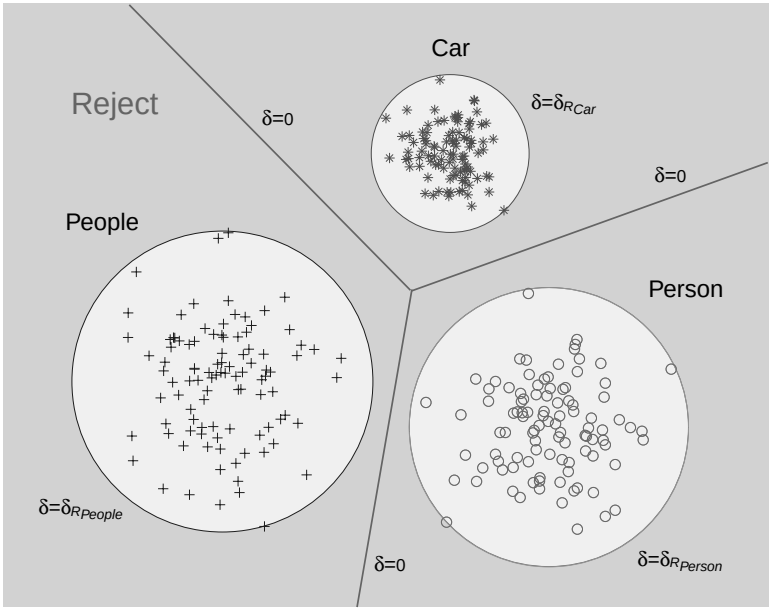


Figure 3.7: Sample partitions before and after the definition of the rejection region

is a problem that has been discussed recently in several publications in the support vector literature [73, 75, 14]. The general approach followed in these papers focuses on explicitly estimating a minimum volume closed region by solving mathematical programs similar to those presented for support vector learning. Given our concerns about the complexity of nonlinear support vector solutions, such approaches are not appealing for our application. Instead we will first learn a low complexity partition of the feature space using differential learning. Then we will constrain the decision regions by increasing the *discriminant differential thresholds* δ_{R_k} (equation 3.6) until the maximum acceptable fraction of correctly classified examples from each class in the validation set is rejected. Figure 3.7 illustrates a hypothetical partition before and after the definition of the rejection region. In practice, the actual decision boundaries will not represent the support of the class-conditional densities as accurately as depicted in the figure. This is not our goal. Our objective is to achieve a balance between computational complexity, classification and rejection performance that meets our objectives. The choice of representation and hypothesis class will determine the degree of our success in terms of classification and rejection performance when using a low complexity partition.

3.6.3 Ranking Image Sequences Based on the Discriminant Differential

Although we are unable to estimate the probability of correct classification reliably, the need remains for some measure of confidence to rank the image sequences collected by the surveillance system. Within the class label distribution space, it is possible to estimate the class-conditional densities $\rho(D|\omega_k)$ in order to rank image sequences based on the maximum likelihood $\rho(D|\omega_*)$. Yet we have chosen to learn a partition of the class label distribution space directly with differential learning to maximize the classification performance of the surveillance system even when limited training data is available.

Many authors [32, 71, 13] have suggested that the margin provides a useful measure of classification confidence; therefore we will consider the value of the discriminant differential as a measure of confidence. The differential would seem to provide a natural measure

Figure 3.8: Contours of constant discriminant differential induced by a linear classifier

of confidence given the data-dependent generalization bounds which indicate that margin maximization leads to improvements in generalization performance. Shawe-Taylor considered the utility of the margin for confidence assessment in [78]. He developed generalization bounds showing that a test example is classified with confidence when a large number of training examples produce margins that are close to the margin of the test example. Avoiding the complexities of the generalization theory presented in [78], we can obtain a qualitatively similar measure of confidence by simply estimating the class-conditional differential densities $\rho(\delta_k(D)|\omega_k)$ and computing the maximum likelihood $\rho(\delta_*(D)|\omega_*)$. Furthermore, if the maximum likelihood $\rho(\delta_*(D)|\omega_*)$ increases monotonically with respect to δ_* for all possible class labels ω_*, all image sequences labeled with a given class label ω_* can be ranked directly based on the differential δ_*.

If we are to efficiently detect unknown object classes and novel views of known object classes by ranking the image sequences based on either the maximum likelihood $\rho(\delta_*(D)|\omega_*)$ or the differential δ_*, we must consider the shape of the surfaces of constant discriminant differential induced by the classifier in the class label distribution space. Since the set of class label distributions $\mathcal{D}_{\delta=\delta_*}$ that produce the differential δ_* all yield the same likelihood $\rho(\delta_*|\omega_*)$, we are unable to discriminate between class label distributions that are elements of the set $\mathcal{D}_{\delta=\delta_*}$. Therefore we would like the surfaces of constant differential to be *closed within the subset of the feature space that the data lies in.*

To emphasize this point, consider the contours of constant discriminant differential induced by a linear and a radial basis function classifier for a two class problem as shown in figures 3.8 and 3.9. The contours of constant discriminant differential for the linear classifier are lines that are parallel to the decision boundary. Due to the fact that the contours are not localized in a region of feature space, the novelty detection performance of the classifier may be disappointing, depending on the nature of the underlying distribution. In cases such as the one illustrated in figure 3.8, we are clearly unable to effectively discriminate between the unknown and known class even though they are separable. In contrast, the contours of constant discriminant differential shown in figure 3.9 do form closed regions about the

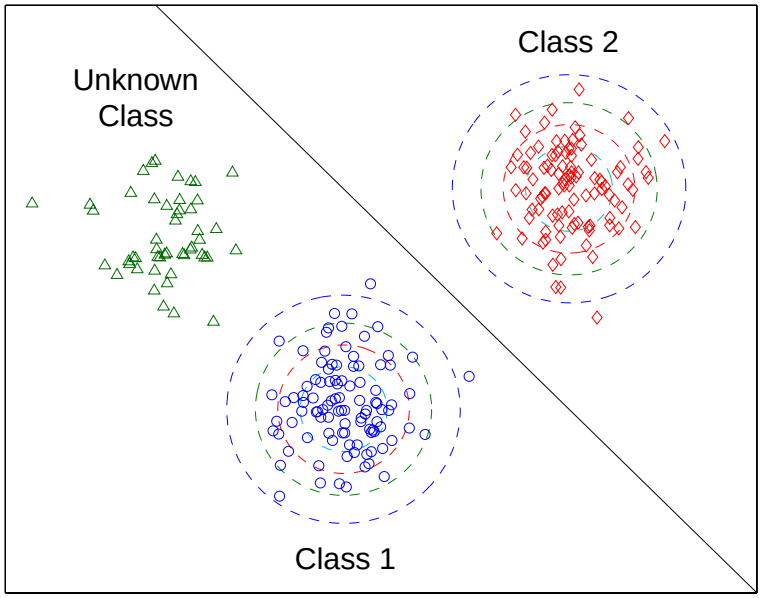


Figure 3.9: Contours of constant discriminant differential induced by a radial basis function classifier

region of feature space containing the training samples. Therefore the radial basis function classifier offers significant improvements in performance in this scenario.

It is important to stress that radial basis function classifiers are not required to perform reliable novelty detection as suggested by some papers in the literature [30, 64]. Whether or not the resulting decision regions are closed is determined not only by the shape of the surfaces of constant differential but also by the nature of the subset of feature space containing the data. For example, if we know that the data is distributed over a unit sphere, a linear classifier is capable of generating localized decision regions on the surface of the sphere. Therefore in cases where we have knowledge of the data space, we may choose to utilize hypothesis classes that do not induce closed decision regions directly in order to simplify the learning process.

3.7 Conclusions

The main objective of this chapter has been to address the problem of learning a partition of a given feature space with a low probability of error. In order to learn a partition that generalizes well to unseen examples, one must perform a tradeoff between minimizing the training error and controlling the capacity of the hypothesis class. Typically, the capacity is assumed to be directly related to the number of classifier parameters; therefore a variety of techniques are traditionally employed to reduce the number of parameters.

Recent experimental and theoretical findings from the statistical learning community indicate that reducing the number of classifier parameters is not a prerequisite to achieve improvements in generalization. For large margin classification techniques, bounds on the generalization performance exist that are *independent of the dimensionality of the feature space*. This suggests that techniques for large margin classification may provide the capability to construct partitions of high dimensional spaces that offer effective classification and rejection performance.

The problem with the widely studied techniques of support vector learning and boosting

is that the resulting classifiers are often quite complex. This precludes the use of these techniques for the design of real-time classifiers. Differential learning provides another option that allows direct control of the complexity of the partition. Since the hypothesis class must be specified prior to learning, we can explicitly constrain our search for large differential partitions to hypothesis classes with low computational complexity.

Modifications to differential learning are needed to address the problems of learning with dependent image data and unknown class prior probabilities. We will compute the sequence image error rate to evaluate the performance of the image classifier. We will select classifiers based on an upper bound on the maximum class-conditional error rate. By selecting the classifier that minimizes this bound, we obtain a partition with the minimum worst case performance.

Evaluating classification confidence is also complicated by the nonstationarity of the problem. Since the training sample is not representative of the underlying distribution, it is impossible to reliably estimate the probability of correct classification for a given example. Therefore other approaches are required for defining the rejection region in image space and rank ordering the image sequences. We introduced techniques based on the differential to address these problems and discussed several design issues.

Chapter 4

Image Classification

4.1 Overview

In the previous two chapters, we focused on defining the elements of the classification process and the design principles for realizing the image and class label distribution classifiers. In the following chapters, we will apply this methodology to a sample classification task and determine whether the resulting classification process supports effective image sequence classification, novel image sequence detection and incremental learning. We begin by performing a series of image classification experiments to explore whether we can successfully design a low complexity classifier that yields excellent image classification and rejection performance for a relevant surveillance task.

4.2 The Classification Task

We will classify image sequences as either individuals, groups of people or cars. The composition of the available dataset for training, cross-validation and testing is detailed in table 4.1 and figure 4.1. These image sequences were obtained by manually associating images from several data collections around the Carnegie Mellon campus. The data collections took place at different times of day under varied lighting conditions. Images containing only portions of moving objects were not included in the dataset since our goal is to reject such examples. The set of false alarm images of moving foliage was used to evaluate rejection performance.

4.3 Classifier Definition and Evaluation

In order to define the image classifier, we need to specify the resolution of the input imagery and the hypothesis class. Our goal is to identify a resolution and hypothesis class that offer excellent classification and rejection performance with minimal computational burden. Since the image sequence data is composed of dependent samples, we will evaluate the classification performance of candidate image classifiers over a series of partitions of the dataset. Each set of experiments performed on a given partition is referred to as a *learning trial*. During the beginning of each learning trial, the available set of image sequences is partitioned into training, validation and test sets. A specified fraction of the total number of images from each class is approximately allocated to each dataset by randomly selecting sequences until the number of images needed for the given dataset is met or exceeded. Half of the dataset is allocated to the training set. The remaining half is split evenly between

	Person	*People*	*Car*	*Foliage*
Images	3965	1303	3877	302
Sequences	306	395	555	-

Table 4.1: Composition of the dataset

the validation and test sets. Fifty partitions of the dataset are randomly selected and used in all of the experiments.

Once the datasets are constructed, learning commences. Five training cycles are executed during a learning trial with different random initializations of the classifier parameters. During each training cycle, backpropagation is utilized to identify classifier parameters which maximize the sample average CFM over the training examples. Momentum is employed to accelerate the learning process. Weight decay is used to regulate the capacity of the hypothesis class. To avoid overfitting, cross-validation is performed during each epoch. The classifier parameters that produce the minimum upper bound on the worst case sequence image error rate are saved. The corresponding discriminant differential thresholds are determined by advancing the threshold for each class until the fraction of correctly classified examples rejected on the validation set equals five percent.

After the candidate classifiers are trained and tested on the fifty partitions of the dataset, the classification and rejection performance of the candidate classifiers are compared by examining the differences in performance for each set of classifiers trained on a given partition of the dataset. The classification performance is compared by examining the differences in the upper bounds on the worst case sequence image error rates on the validation sets. The rejection performance is compared by examining the differences in the rejection rates on the set of images of moving foliage. Once a candidate classifier is selected, the classifier which yields the minimum worst case sequence image error rate bound in cross-validation over the fifty learning trials is evaluated in detail.

4.4 The Logistic Linear Classifier

To minimize the computational complexity initially, we will use the logistic linear classifier as our baseline image classifier. The logistic linear classifier consists of a set of $\mathcal{C}$ logistic linear discriminant functions of the form

$$g_i(X|\theta_i, \theta_{b_i}) = f(\theta_i^T X + \theta_{b_i}) \tag{4.1}$$

where

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

and X is the vectorized $N \times N$ image. To determine the class label, the largest discriminant function output $g_{(1)}(X|\theta_{(1)}, \theta_{b_{(1)}})$ is identified and the discriminant differential is computed for the hypothesized class. If the discriminant differential exceeds the threshold associated with the class, then the class label $\omega_{(1)}$ is assigned to the image; otherwise the image is rejected.

Let us examine what types of surfaces of constant discriminant differential are induced by the logistic linear classifier. Consider the example in figure 4.2. The solid curves are the contours of constant discriminant differential formed by the largest discriminant function $g_{(1)}(X|\theta_{(1)})$ and the next largest discriminant function $g_{(2)}(X|\theta_{(2)})$. The dashed lines are the lines

$$\theta_{(1)}^T X + \theta_{b_{(1)}} = 0 \tag{4.3}$$

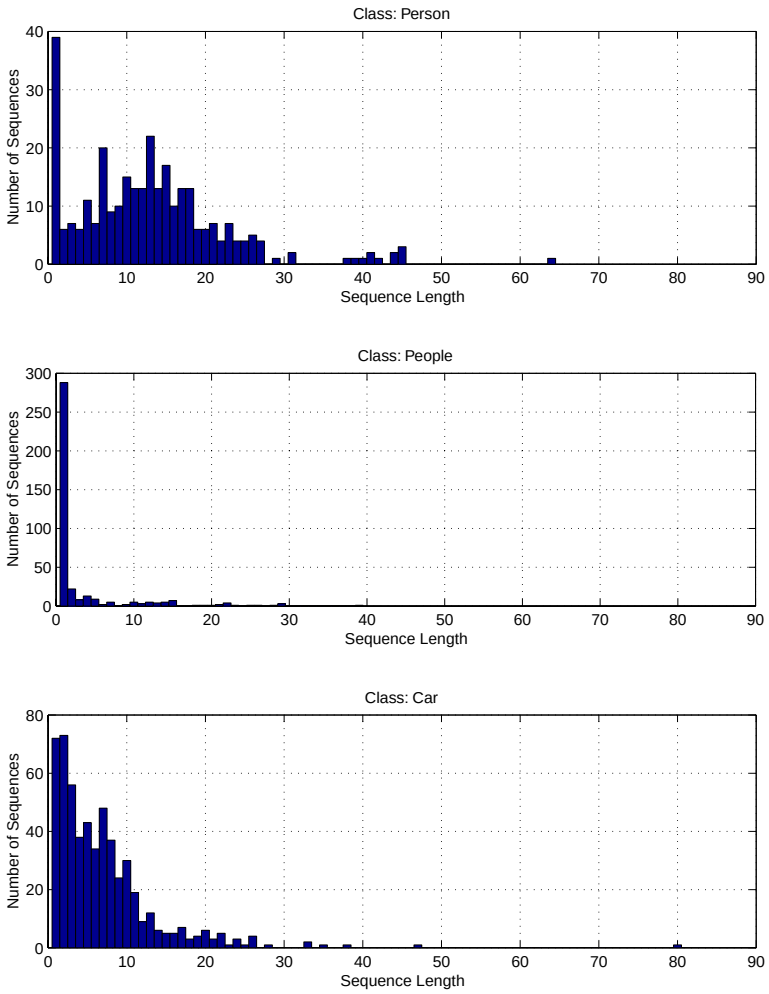


Figure 4.1: Class sequence length distributions for the dataset

and

$$\theta_{(2)}^T X + \theta_{b_{(2)}} = 0, \tag{4.4}$$

perpendicular to the corresponding weight vectors $\theta_{(1)}$ and $\theta_{(2)}$, along which the logistic linear discriminant functions' rates of change are maximum. We will refer to these lines as the *principal lines*.

As one moves along a contour of constant discriminant differential away from the principal line intersection, the contour becomes parallel to the nearest principal line. This indicates that the rate of change of the corresponding discriminant function becomes the dominant term in the rate of change of the discriminant differential. In general, we can show that for any arbitrary pair of logistic linear discriminant functions, the surfaces of constant discriminant differential become parallel to the nearest *principal hyperplane* as one moves along a surface of constant discriminant differential away from the principal hyperplane intersection (see appendix C for details). Therefore as figure 4.2 suggests, the logistic linear classifier defines wedges in image space that are mapped to the various object classes.

Since the logistic linear classifier will be used to partition the hypercube $[0, 255]^{N^2}$, one may suspect that the rejection performance of the logistic linear classifier will not be ideal due to the fact that the classifier is unable to generate closed surfaces of constant

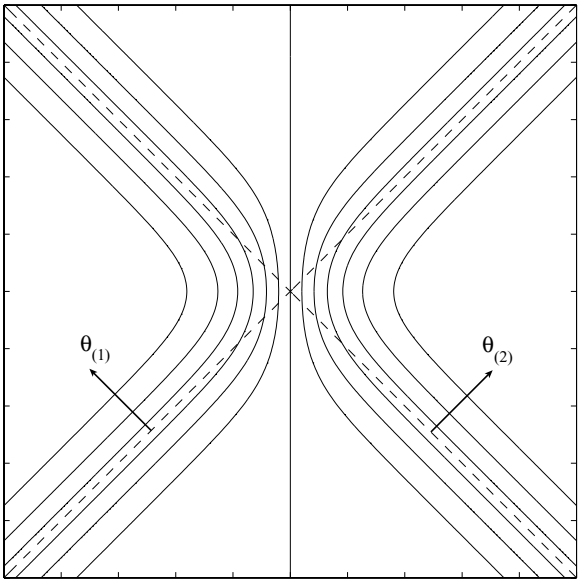


Figure 4.2: Contours of constant discriminant differential generated by two logistic linear discriminant functions

discriminant differential in the data space. As we shall see in the following experiments, the key to maximizing the rejection performance of the logistic linear classifier lies in the minimization of the variance of the class-conditional discriminant differential densities. By minimizing the variance, the discriminant differential thresholds can be increased without rejecting a significant fraction of the correctly classified examples.

4.5 Logistic Linear Image Classification and Rejection

4.5.1 The Baseline Image Classifier

In order to investigate the classification and rejection performance of the logistic linear classifier as a function of the image resolution, we have run a series of experiments using 20×20, 30×30 and 40×40 pixel imagery. Figure 4.3(a) presents box plots [80, Ch. 2] of the sequence image error rate bounds on the validation sets for the fifty learning trials. Figure 4.3(b) presents box plots of the reductions in the bound obtained from incremental increases in the image resolution. Based on these figures, it appears that there are no significant differences between the classifiers in terms of classification performance. The classifier processing 30×30 pixel imagery provides only marginal improvements in performance on average over the classifier processing 20×20 pixel imagery. Increasing the resolution beyond 30×30 pixels slightly degrades the performance of the classifier on average.

To understand the degradation in performance caused by the transition from 30×30 to 40×40 pixel imagery, consider the box plots in figure 4.4 of the worst case sequence image error rate bounds when rejection is not permitted. As we would have hoped, maximizing the differentials effectively controls the capacity of the hypothesis class and delivers improvements in classification performance as the image resolution is increased. Yet at the same time, the variance of the class-conditional discriminant differential densities must be increasing in order for the 40×40 pixel classifier to lag behind when rejection is permitted. As the variance increases, the discriminant differential thresholds must be reduced in order to avoid rejecting a larger fraction of the correctly classified examples. This will subse-

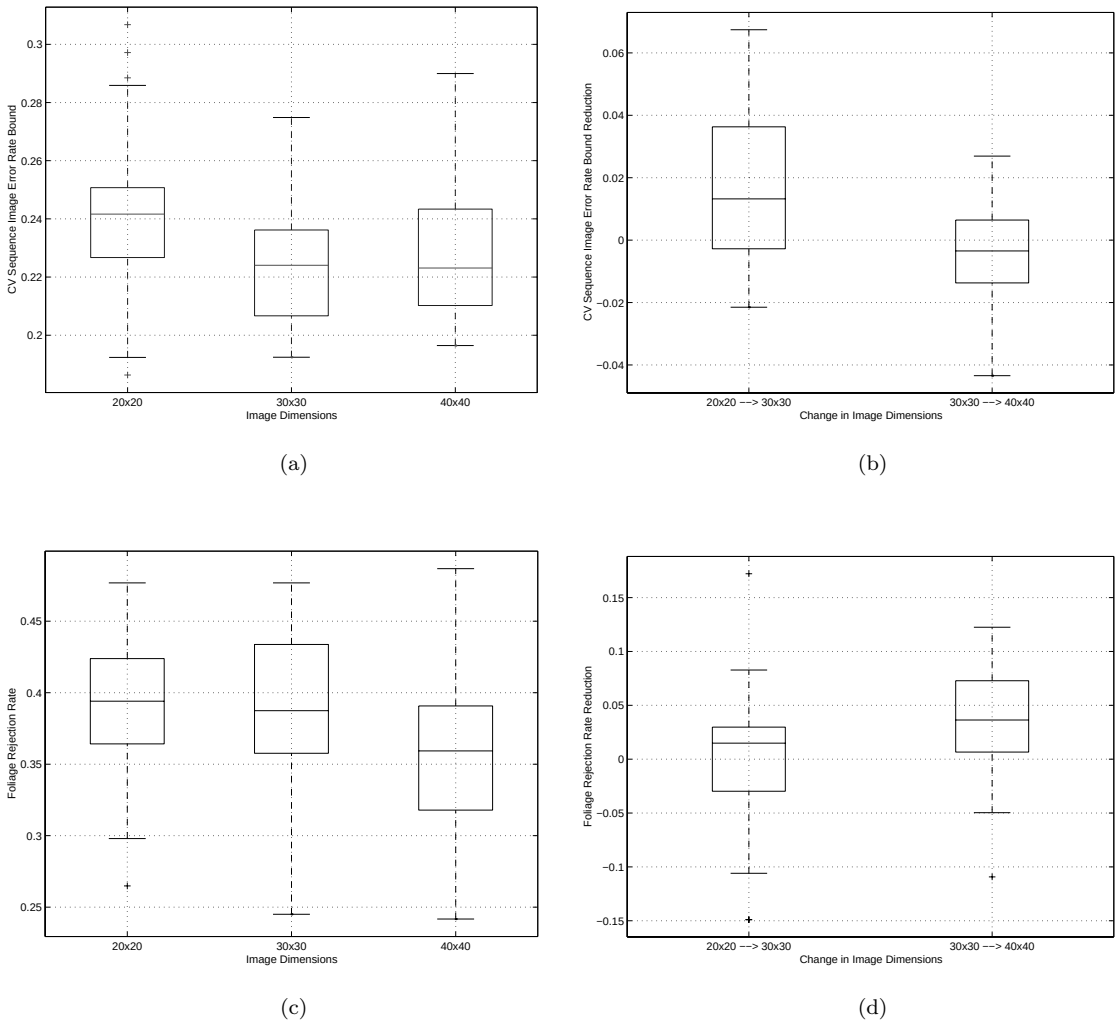


Figure 4.3: Logistic linear classification and rejection performance: (a) Box plots of the sequence image error rate bounds on the validation sets for multiple image resolutions (b) Box plots of the reduction in the sequence image error rate bounds for incremental increases in image resolution (c) Box plots of the foliage rejection rates for multiple image resolutions (d) Box plots of the reduction in the foliage rejection rates for incremental increases in image resolution

quently cause increases in the error rate bounds and reductions in the foliage rejection rates as observed in figures 4.3(b) and (d).

Considering the rejection performance of the logistic linear classifiers now, we discover rather disappointing results. Figure 4.3(c) presents box plots of the foliage rejection rates for the fifty learning trials. Figure 4.3(d) presents box plots of the reductions in the rejection rate obtained after incremental increases in the image resolution. At all resolutions, the logistic linear classifier was unable to reject a majority of the foliage false alarms. As the image resolution increased, the foliage rejection rate generally decreased.

In order to gain more insight into the performance of the logistic linear classifier, we examined the 30×30 pixel image classifier yielding the minimum error rate bound on the validation set in detail. After reviewing the test sequences containing misclassified images, we discovered the majority of the errors fall into two categories. Two examples of each type of error are presented in figure 4.5. The most common type of error involves images of a person who is not centered within the image chip. This is often caused by a cast shadow. The other type of error is caused by images of pairs of people where one person is occluding another person who generally is poorly illuminated. For such examples, the class label *person* was assigned to the training examples if less than half of the occluded person was visible.

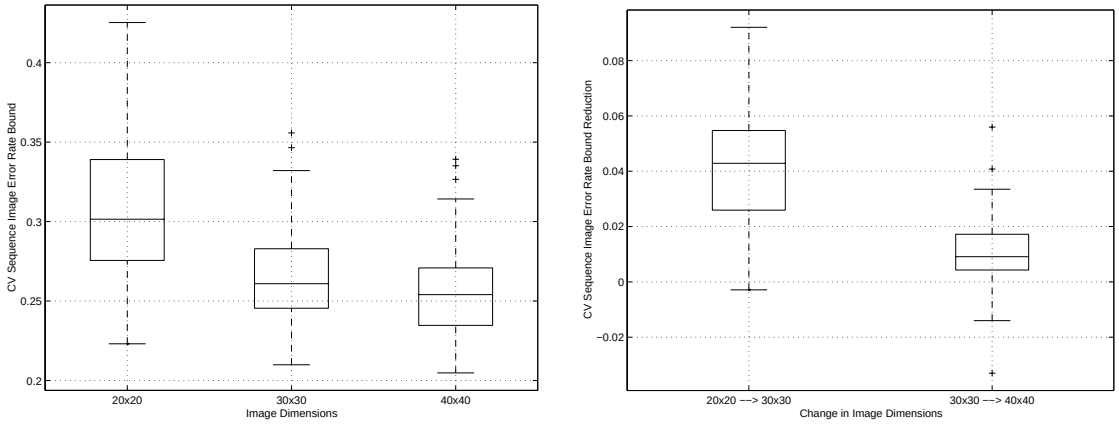
To understand the poor performance in foliage rejection, one should examine the classifier weights for the 30×30 pixel classifier shown in figure 4.6. The weight layer associated with the *people* discriminant function is quite complex and unstructured. This is not surprising given that people walking together can be observed in a variety of configurations relative to one another. The consequences of such a complex representation can be seen in the class-conditional discriminant differential densities for the people and foliage test examples shown in figures 4.7 and 4.8. The lack of structure in the weights leads to a long tail in the discriminant differential density for the people class. Therefore we are unable to set the threshold for the people class very high without rejecting a significant fraction of the correctly classified examples. This in turn leads to poor rejection performance on the foliage examples. Only 39% of the foliage images are rejected. 70% of the foliage false alarms are assigned to the people class.

To achieve improvements in rejection performance, we need to reduce the variance in the class-conditional discriminant differential densities. There are two directions we can pursue to attain this goal. One option is to a select a new hypothesis class which allows us to construct more complex partitions of the image space. Another option is to modify the image representation in an attempt to reduce the intra-class variance in feature space. Ultimately, we want to minimize the overall computational complexity of the representation and the partition. We believe modifications to the image representation may allow us to achieve this objective. Therefore we will consider two image normalization procedures for achieving differential variance reduction and reducing the number of errors caused by random image translations.

4.5.2 Agnostic Image Normalization

In order to reduce the variance of the data, designers typically perform one or several preprocessing steps prior to classification. When processing images, such preprocessing steps often include mean removal, normalizing the magnitude of the image and centering the image with respect to the image's center of mass. We will examine the effect of horizontally centering the images on the performance of the classifier.

Figure 4.9 illustrates the classification and rejection performance of the logistic linear classifier when processing centered images. Once again the classifier processing 30×30 pixel imagery provides a slight advantage over the other classifiers on average. Figure 4.10



(a) (b)

Figure 4.4: Logistic linear classification performance with no rejection: (a) Box plots of the sequence image error rate bounds on the validation sets for multiple image resolutions (b) Box plots of the reduction in the sequence image error rate bounds for incremental increases in image resolution

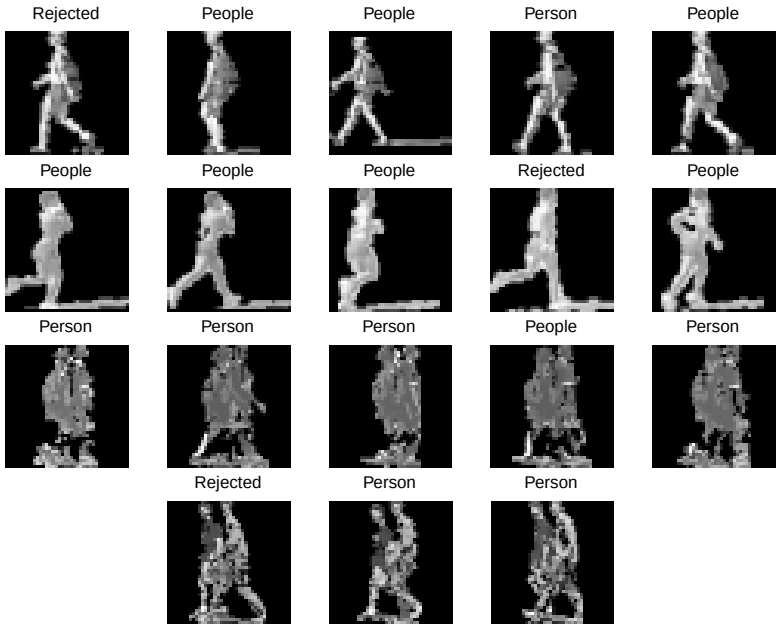


Figure 4.5: Portions of image sequences classified by the logistic linear classifier containing errors

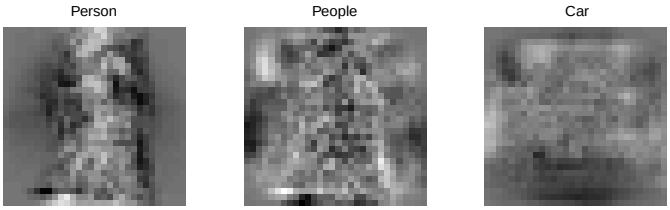


Figure 4.6: Classifier weights for the 30×30 pixel logistic linear classifier (Light regions correspond to positive weights and dark regions correspond to negative weights)

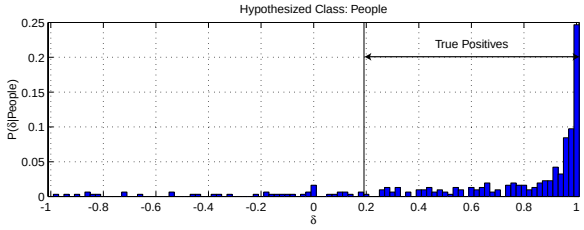


Figure 4.7: Discriminant differential density for 30×30 pixel images of people classified by the logistic linear classifier (The black vertical line denotes the rejection threshold)

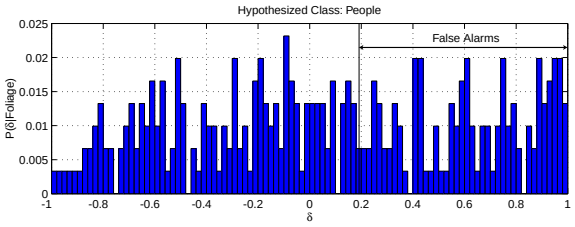


Figure 4.8: Discriminant differential density for 30×30 pixel images of foliage classified by the logistic linear classifier assuming people is the correct class (The black vertical line denotes the rejection threshold)

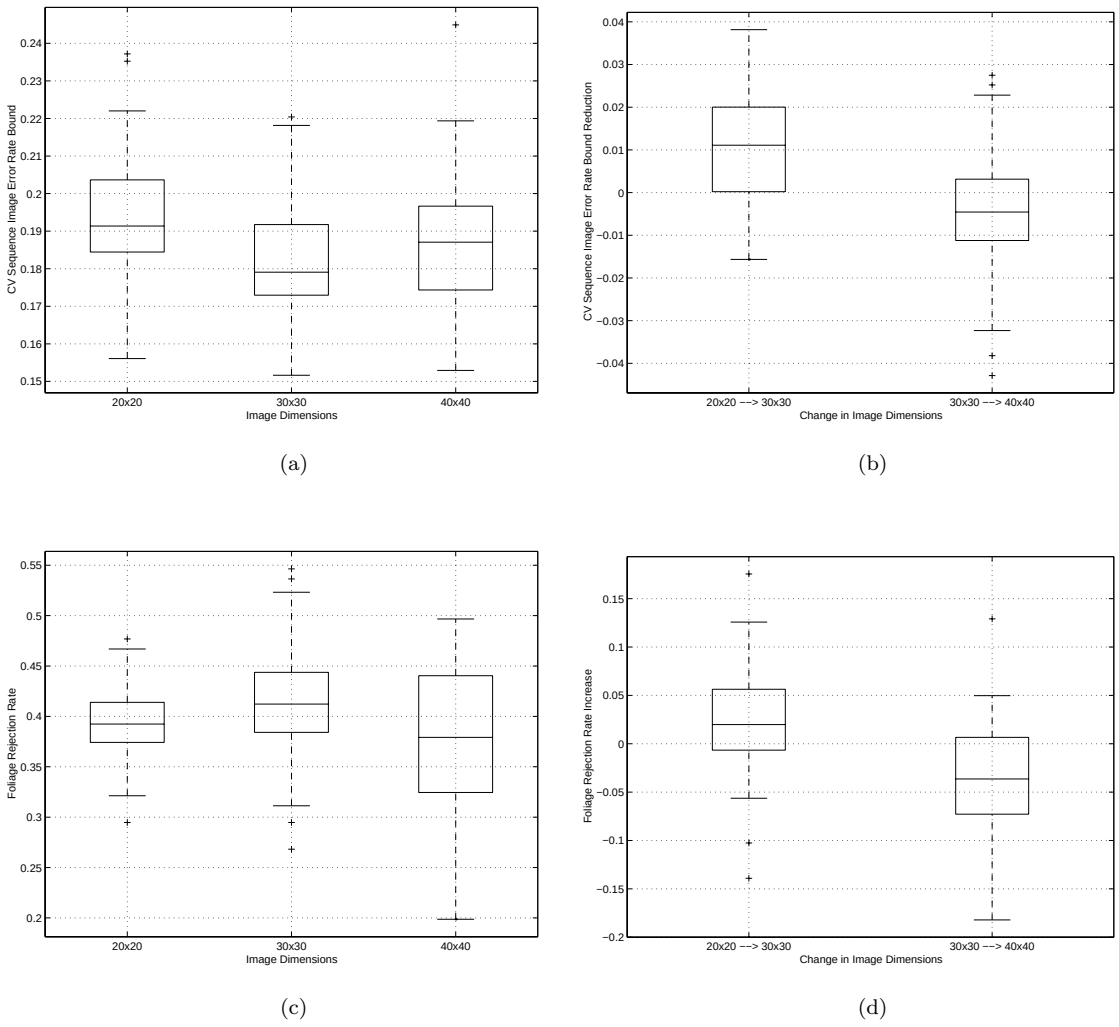


Figure 4.9: Logistic linear classification and rejection performance when using centered images: (a) Box plots of the sequence image error rate bounds on the validation sets for multiple image resolutions (b) Box plots of the reduction in the sequence image error rate bounds for incremental increases in image resolution (c) Box plots of the foliage rejection rates for multiple image resolutions (d) Box plots of the reduction in the foliage rejection rates for incremental increases in image resolution

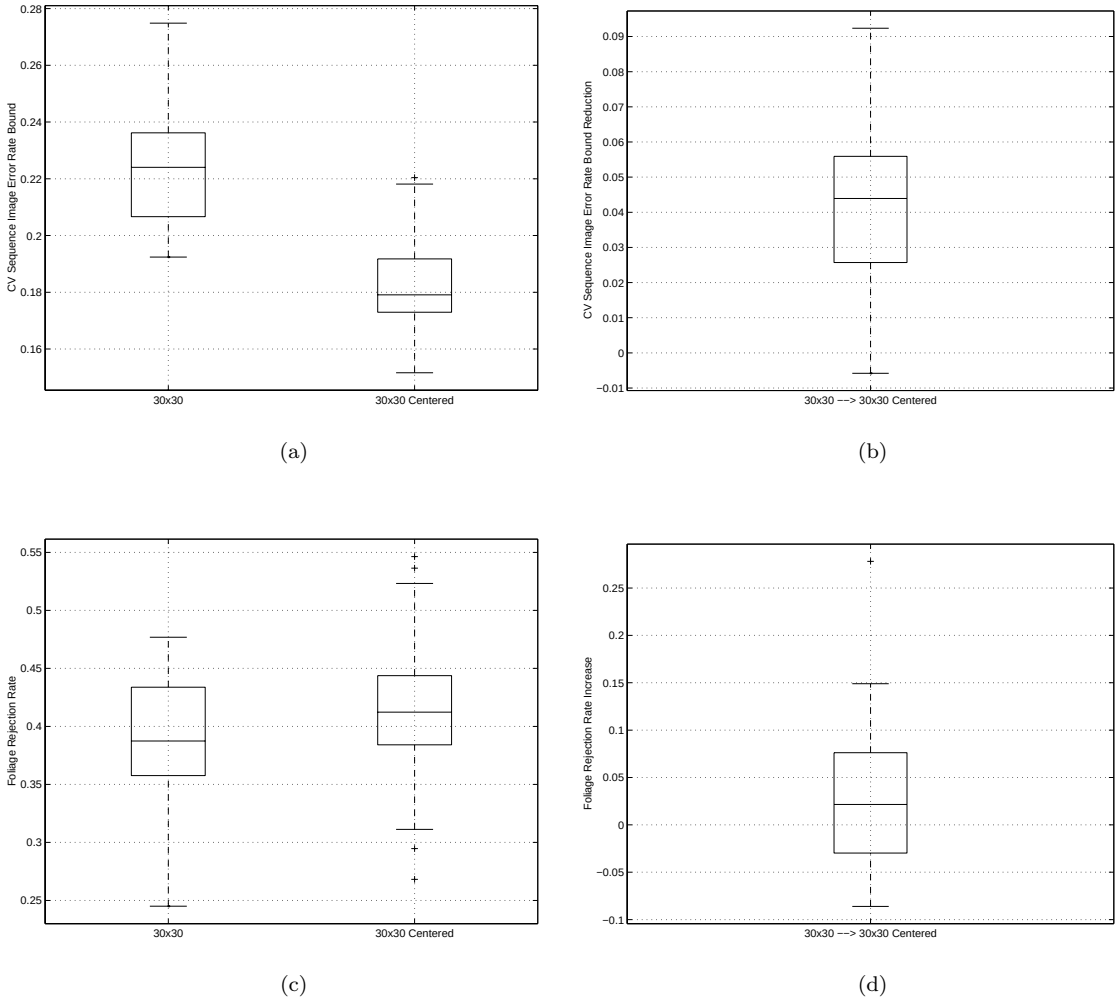


Figure 4.10: The effect of image centering on classification and rejection performance: (a) Box plots of the sequence image error rate bounds on the validation sets before and after centering (b) Box plot of the reduction in the sequence image error rate bounds (c) Box plots of the foliage rejection rates before and after centering (d) Box plot of the increase in the foliage rejection rates

compares the logistic linear classifiers processing 30×30 pixel centered and uncentered imagery. Processing centered imagery yields improvements in classification and rejection performance. Yet the rejection performance is still far from ideal.

Let us now consider the 30×30 pixel image classifier producing the minimum error rate bound on the validation set. Examining the classifier weights in figure 4.11, we see that the complexity remains in the weights of the *people* discriminant function. Not surprisingly, we discover the foliage rejection performance is essentially unchanged from the previous classifier. 40% of the foliage images are rejected. 75% of the foliage false alarms are assigned to the *people* class.

Centering the images clearly does not offer an effective solution for reducing the variance of the class-conditional discriminant differential densities. Since the normalization procedure was selected without regard for the classification task, this is not surprising. By choosing to normalize the images with respect to the center of mass, we have implicitly chosen a configuration for the training examples in the image space. Prior to training and testing the image classifier, we do not know whether our choice will help or hinder our cause. Ideally, we would like to incorporate the normalization process into the scope of the learning process so that the normalization procedure can be selected in a principled manner with respect to the objective function. We introduce a procedure in the next section which allows us to achieve this goal.

4.5.3 Learning to Normalize the Images

In order to minimize the variance of the class-conditional densities directly, we must couple the image normalization process and the classifier in some manner. Instead of employing an agnostic normalization procedure which normalizes a given image based on a property of the image, we will normalize the image such that we *maximize the discriminant differential of the translated image*. This implies that for a given image I, we will compute the discriminant differentials

$$\delta(\mathcal{T}(I,n)|\theta) = g_{(1)}(\mathcal{T}(I,n)|\theta) - g_{(2)}(\mathcal{T}(I,n)|\theta) \tag{4.5}$$

for a range of translations of the image $\mathcal{T}(I,n)$. Once we have identified the translation n_{max} which maximizes the discriminant differential, we will transform the original image I to $\mathcal{T}(I,n_{max})$ so that

$$\delta(\mathcal{T}(I,n_{max})|\theta) = \max_n \delta(\mathcal{T}(I,n)|\theta). \tag{4.6}$$

Through this procedure, we are encouraging large differentials for all of the examples; this eliminates the long tails in the densities that constrain the size of the rejection region. Given that the discriminant functions determine how the examples in image space will be normalized, we are implicitly learning the normalization process when learning the partition.

Figure 4.12 presents the classification and rejection results obtained when the images are normalized by maximizing the discriminant differential. The 20×20 pixel image classifier translates images up to 10 pixels in either direction in 1 pixel increments. The 30×30 pixel image classifier translates images up to 14 pixels in either direction in 2 pixel increments. The 40×40 pixel image classifier translates images up to 18 pixels in either direction in 3 pixel increments. Figures 4.12(a) and (b) indicate there is no significant difference in classification performance between the classifiers on average. The classifiers processing 20×20 and 30×30 pixel imagery deliver essentially equivalent classification performance. Figures 4.12(c) and (d) demonstrate a fairly consistent decrease in the foliage rejection rate on average as the image resolution increases. Therefore the 20×20 pixel image classifier provides advantages over the other classifiers both in terms of rejection performance *and* computational complexity.

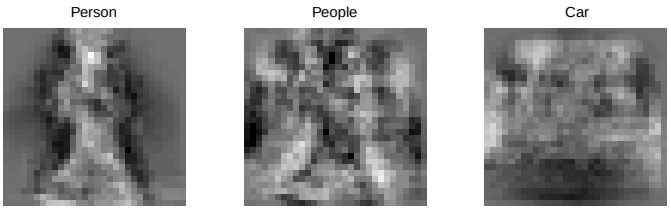
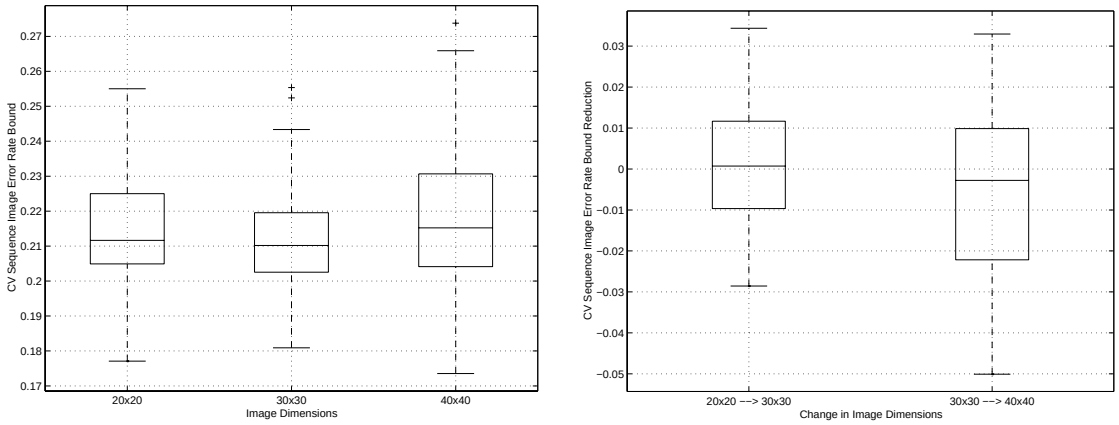
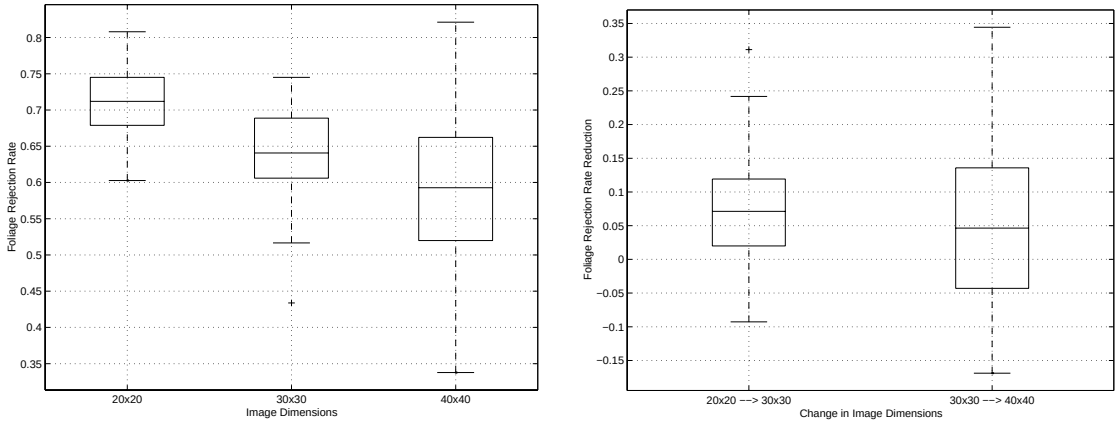


Figure 4.11: Classifier weights for the logistic linear classifier processing 30×30 pixel centered images (Light regions correspond to positive weights and dark regions correspond to negative weights)



(a) (b)



(c) (d)

Figure 4.12: Logistic linear classification and rejection performance when normalizing based on the discriminant differential: (a) Box plots of the sequence image error rate bounds on the validation sets for multiple image resolutions (b) Box plots of the reduction in the sequence image error rate bounds for incremental increases in image resolution (c) Box plots of the foliage rejection rates for multiple image resolutions (d) Box plots of the reduction in the foliage rejection rates for incremental increases in image resolution

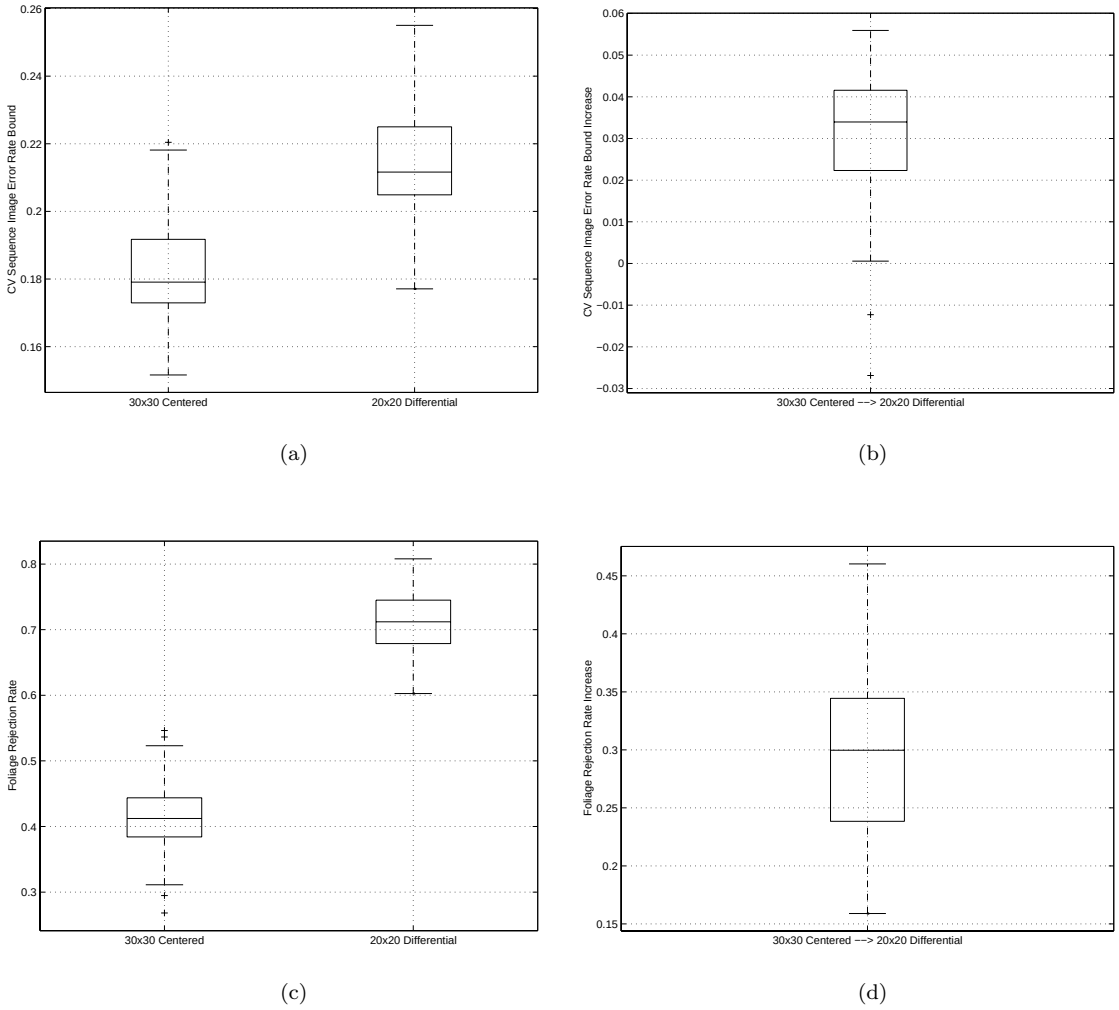


Figure 4.13: Image centering versus normalization based on the discriminant differential: (a) Box plots of the sequence image error rate bounds on the validation sets (b) Box plot of the increase in the sequence image error rate bounds when normalizing based on the discriminant differential (c) Box plots of the foliage rejection rate (d) Box plot of the increase in the foliage rejection rates when normalizing based on the discriminant differential

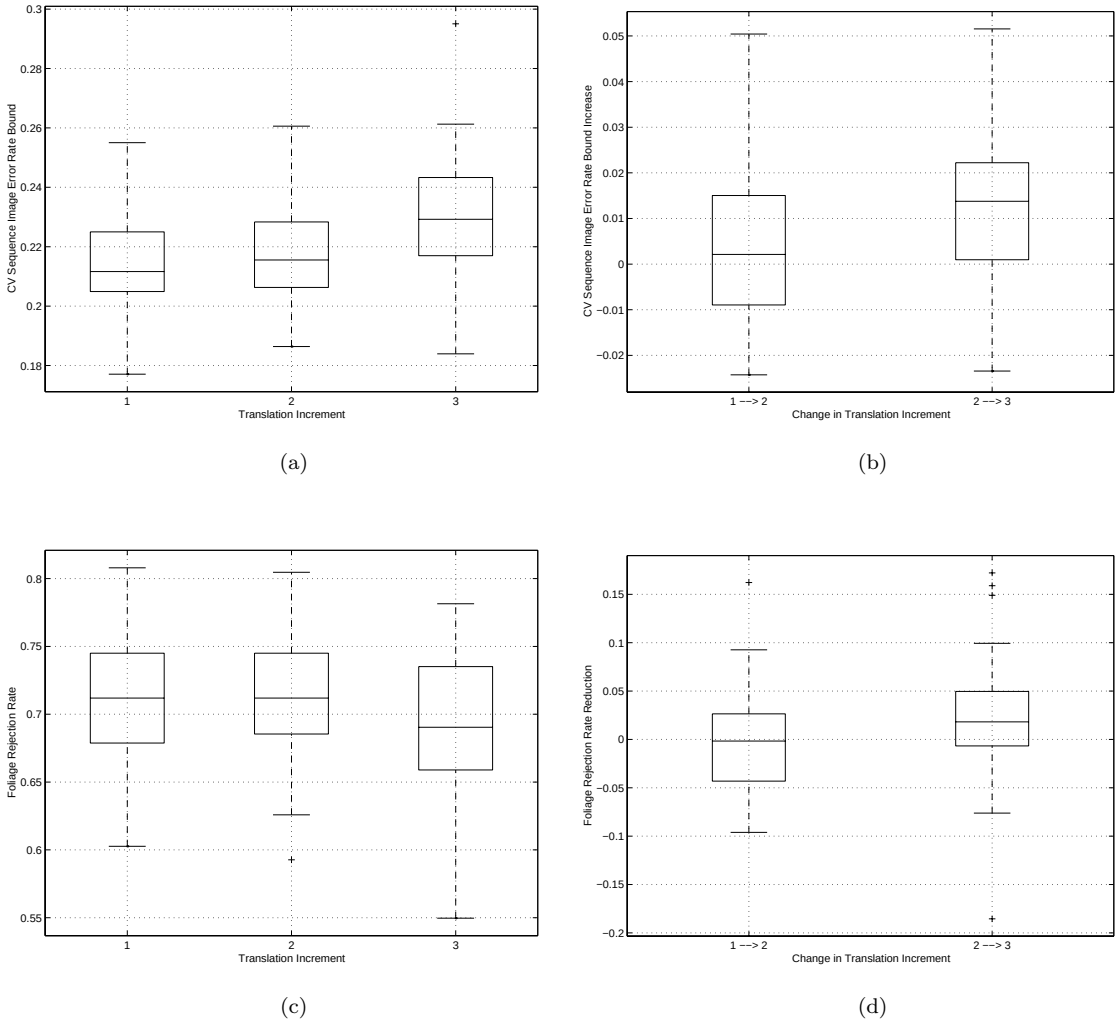


Figure 4.14: The effect of varying the translation increment: (a) Box plots of the sequence image error rate bounds on the validation sets (b) Box plots of the increase in the sequence image error rate bounds when increasing the translation increment (c) Box plots of the foliage rejection rates (d) Box plots of the reduction in the foliage rejection rates when increasing the translation increment

We now compare the 20×20 pixel image classifier with the 30×30 pixel image classifier processing centered images. Figure 4.13 illustrates their relative performance. In terms of classification performance, the 20×20 pixel image classifier consistently lags behind the 30×30 pixel image classifier by several percent over the fifty learning trials. Yet the 20×20 pixel image classifier establishes a new standard in rejection performance by providing nearly a factor of two improvement in the median rejection rate. This performance improvement obviously comes at a price. Aside from the minor reduction in classification performance, the computational cost of evaluating the classifier has increased by nearly an order of magnitude. To reduce the computational burden, we investigated the effect of varying the translation increment. Figure 4.14 shows the degradation in performance as the translation increment is increased. It appears that the translation increment can be increased to 2 pixels without a significant penalty in classification and rejection performance. We will now attempt to obtain improvements in performance by modifying the image representation further.

4.5.4 The Role of the Intensity Data

When presented with an image of a moving object, we are given two types of information about the object on which to base our classification decision. First of all, the image provides a representation of the radiance [41] from the visible object surfaces. In addition, if the image intensities are all greater than zero, the image implicitly provides a representation of the object's shape. Given the variability in outdoor lighting conditions and limited image resolution, it is not clear that the intensity data will aid in discriminating between individuals, groups of people and cars. At low resolution, it will be difficult to discern specific object features. Therefore by reducing the original image to a binary image representing the object shape, we can reduce the intraclass variance while preserving the discriminative features of the images.

One of the additional benefits of processing binary images is that the nature of the binary image space allows the logistic linear classifier to induce closed surfaces of constant discriminant differential. Binary images lie on the surface of the hypercube $[0,1]^{N^2}$. This allows semi-infinite decision surfaces such as hyperplanes to generate localized decision regions in the binary image space.

Comparing the rejection performance of the 20×20 pixel image classifiers processing the original and binarized images in figure 4.15, we find that binarizing the imagery clearly offers a significant advantage. At the same time, binarization does not degrade classification performance on average. Given our success with this classifier, we selected this design for further evaluation on the test data and implementation. The test sample classification and rejection results for the classifier producing the minimum error rate bound on the validation set are listed in tables 4.2 through 4.5. Figure 4.16 presents the classifier weights. Figure 4.18 presents the test sequence image error rate bounds over the fifty trials.

Notice that the classifier weights are clearly more structured than the weights corresponding to the previous classifiers. In each discriminant function, one can now discern a specific region of the weights that corresponds to the feature detector for the given object class. Based on the configuration of the feature detectors, one would surmise that centered examples of individuals will generally remain in the center while examples of groups of people are translated to the right and cars are translated to the left. The translation histograms in figure 4.17, which illustrate how the examples from each class are translated, confirm this conjecture.

The test results seem to indicate that we should expect good generalization performance for known objects similar to those encountered in the dataset. Although the confidence bounds on the class-conditional error rates are large, we expect that the sequence error rate will be reasonable even in the worst case due to the integration of classification results

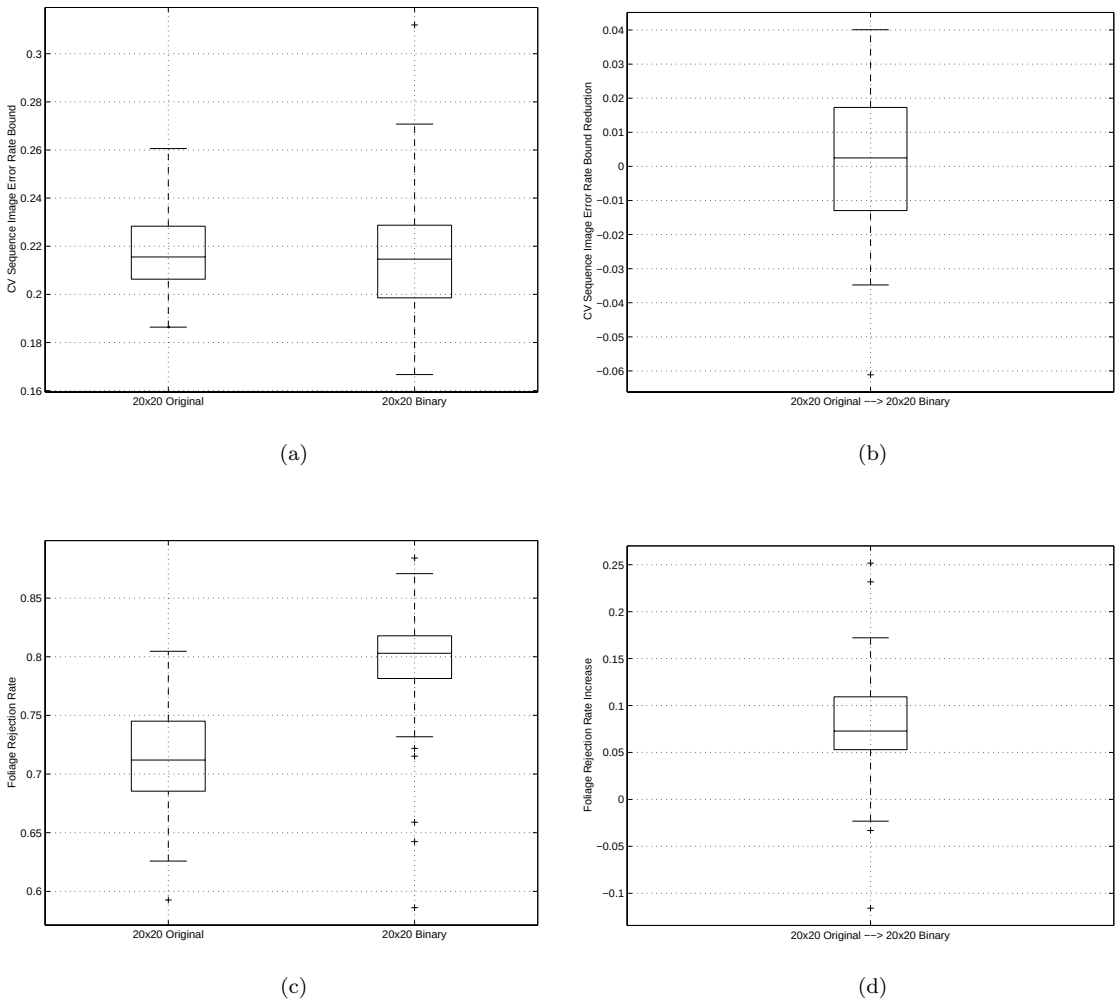


Figure 4.15: The effect of binarizing the images: (a) Box plots of the sequence image error rate bounds on the validation sets (b) Box plot of the reduction in the sequence image error rate bounds after binarizing the images (c) Box plots of the foliage rejection rates (d) Box plot of the increase in the foliage rejection rates after binarizing the images

	Person	*People*	*Car*	*Rejected*
Person	**921**	26	0	31
People	17	**277**	7	24
Car	0	23	**875**	39
Foliage	11	37	10	**244**

Table 4.2: Test image confusion matrix for the 20×20 pixel binary image classifier

Class	Holdout Estimate	95% Confidence Interval
Person	0.028	[0.000,0.198]
People	0.051	[0.000,0.181]
Car	0.016	[0.000,0.128]

Table 4.3: Class-conditional sequence image error rate estimates for the 20×20 pixel binary image classifier

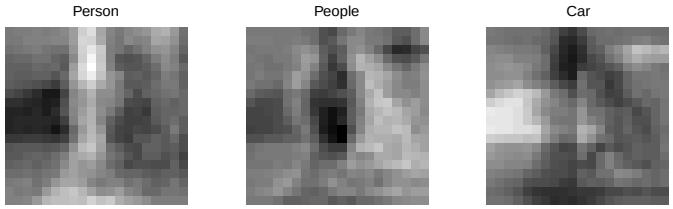


Figure 4.16: Classifier weights for the logistic linear classifier processing 20×20 pixel binary images (Light regions correspond to positive weights and dark regions correspond to negative weights)

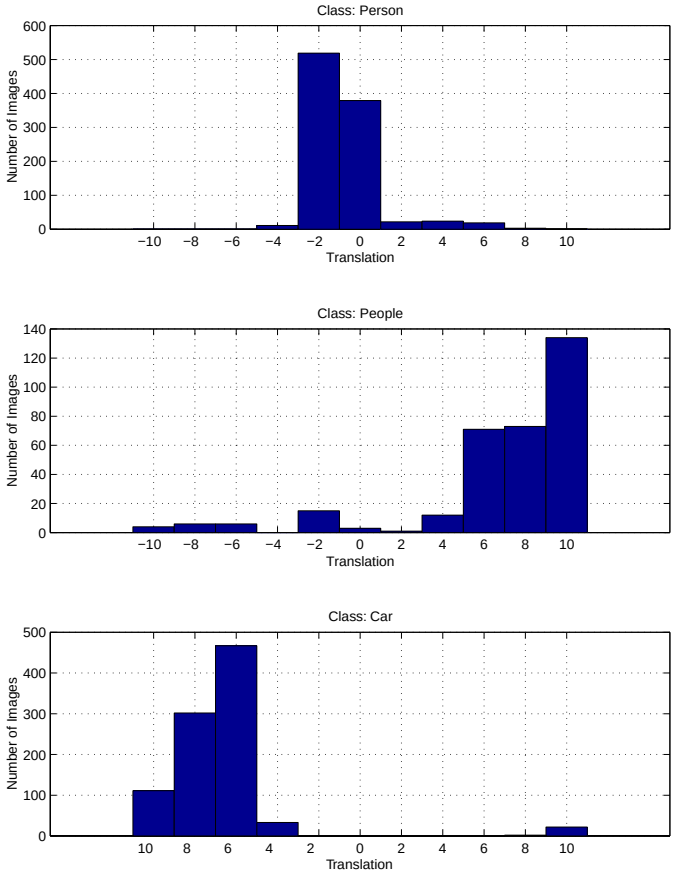


Figure 4.17: Translation histograms for the object classes (Negative translations are to the left and positive translations are to the right)

Class	Holdout Estimate	95% Confidence Interval
Person	0.046	[0.000,0.216]
People	0.115	[0.000,0.245]
Car	0.052	[0.000,0.164]

Table 4.4: Class-conditional sequence image rejection rate estimates for the 20×20 pixel binary image classifier

Class	Holdout Estimate
Foliage	0.81

Table 4.5: Foliage rejection rate estimate for the 20×20 pixel binary image classifier

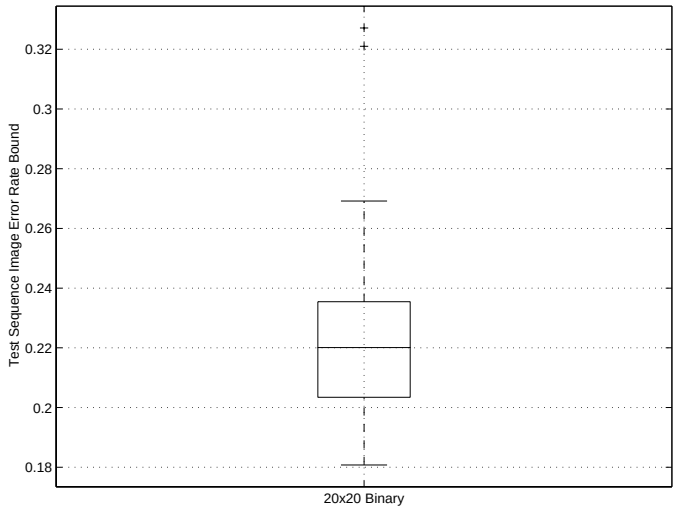


Figure 4.18: Box plot of the test sequence image error rate bounds for the logistic linear classifier processing 20×20 pixel binary images

over time. Based on the foliage rejection performance, we believe the classifier will support novel image sequence detection when the objects present distinctly different shapes from the known objects. Further experimentation is needed to investigate the sensitivity of the classifier to more subtle variations in shape. We will pursue this issue in the next chapter.

4.6 Comparison with Related Classifiers

There are several classifiers presented in the literature for real-time moving object classification. The majority of the classifiers base their decisions on a small number of object properties such as height, area and aspect ratio [12, 31, 47]. These properties are determined after localizing the object in the environment. Given such low dimensional feature spaces, it will be impossible to reliably discriminate between objects with similar dimensions. Therefore we do not expect these classifiers will support effective false alarm rejection and novel image sequence detection. The majority of the papers describing these classifiers do not address the issue of rejection. The CMU/VSAM group provides the only estimate of false alarm rejection performance for their person-people-vehicle classifier. On a set of 48 false alarm examples, the classifier achieves a rejection rate of 64.5%. This leaves a significant margin for improvement.

Foresti [26, 27] has introduced the only other real-time appearance-based classifiers for video surveillance. Each classifier bases the decision on a set of invariant features extracted from a binary image. Since the papers do not clearly describe the nature of the representations used in the experiments, it is difficult to comment on potential rejection performance. No evaluation of rejection performance was presented in either paper.

4.7 Conclusions

From this series of experiments, we have demonstrated that efficient image classification and rejection can be performed using a low complexity classifier in the context of the given classification task. This is an encouraging first step. We will now review these results to identify weaknesses and discuss improvements that will allow us to design low complexity image classifiers for more complex classification problems.

Although the classification problem we addressed here was not particularly challenging, the identification of a suitable combination of image representation and partition was nontrivial. Typically the core issues faced during the design of a pattern classification system are issues of representation. This problem is no exception. To achieve an acceptable balance between classification performance, rejection performance and computational complexity, we selected various representations and evaluated their performance using the available dataset. This can be a long and fruitless search. As the image normalization experiments suggest, it is preferable to pose the learning problem such that representation decisions are made by the learning procedure in the context of the classification problem. We would ultimately like the learning system to search a given class of image representations and partitions for classifiers that provide the appropriate balance between performance and computational complexity. But for now, the designer will play an integral role in the design process.

In our pursuit of variance reduction, we explored one option of modifying the image representation. By coupling the image normalization process with the classifier, we managed to achieve significant improvements in rejection performance. Yet it is not clear in hindsight that normalization based on the discriminant differential offers the most efficient and effective solution. Recall once again the outcome from the learning procedure when normalizing based on the discriminant differential. The logistic linear classifier formed distinct feature

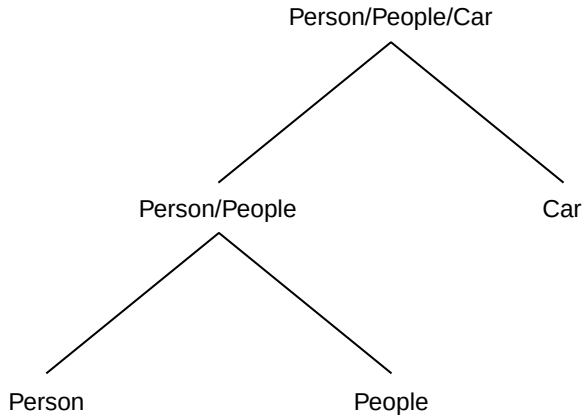


Figure 4.19: Hierarchical binary image classifier

detectors for each object class in different regions of the weights. This allowed the classifier to produce larger discriminant differentials since only one discriminant function will respond when the active portion of the image is contained within the region corresponding to one of the feature detectors.

Another approach that will allow us to achieve a similar end involves adding complexity to the hypothesis class. By employing a multi-layer perceptron classifier instead of the logistic linear classifier, for example, we can construct a larger set of feature detectors that can be used to increase the differentials for the more challenging examples that lie near the decision boundary when the images are normalized based on the center of mass. Yet in order to avoid squandering the computational savings obtained by eliminating the classifier evaluations for multiple translations, we should think carefully about the structure of the classifier. As the classification problems that we address in the future become more complex, it will become increasingly difficult to rapidly evaluate the image classifier and deliver information to the user in real-time. Monolithic classifiers such as multi-layer perceptrons which directly map a given input to one of $\mathcal{C}$ classes will not make this task any easier.

To ease the computational burden while maximizing the flow of information to the user, we believe a hierarchical image classifier will be critical. In a hierarchical classifier, the image is not directly mapped to a given object class. Instead, a series of classification decisions are made which incrementally reduce the list of candidate object classes to the most likely object class. The benefits of decomposing the decision process in this manner are twofold. By incrementally eliminating hypotheses, we reduce the amount of computation required to perform the discrimination. In addition, if the system is unable to complete the decision cycle in time, the system can provide the user with a list of possible class labels. This list can be refined further during continued observation of the object.

To provide a simple example of a hierarchical classifier, we decomposed the 20×20 pixel binary image classifier into a hierarchical classifier composed of two logistic linear classifiers. Figure 4.19 illustrates the decision hierarchy. By decomposing the classifier, we reduce the number of dot products from three to one or two per translation depending on the path taken through the tree. As the complexity of the classifier increases, the computational savings should be even more significant. An interesting line of future research will be to investigate how to automatically construct such a hierarchy.

Chapter 5

Image Sequence Classification and Novelty Detection

5.1 Overview

After defining the image classifier, our next objective is to define the class label distribution classifier and evaluate the performance of the classification process. Using the image classifier, we will generate the collection of class label distributions for the training and test image sequences. Once we have selected a hypothesis class, we will learn a class label distribution classifier from the data and evaluate the image sequence classification performance of the classification process on the test set. Then we will evaluate the ability of the classification process to assist the user in the detection of image sequences of unknown objects.

5.2 The Class Label Distribution Space

In order to simplify our evaluation of the class label distribution classifier, we begin by attempting to visualize the class label distribution space $\mathcal{L}$. The *class label distribution* $D(S)$ for a given image sequence S is a $\mathcal{C}$+1-dimensional vector indicating the fraction of images assigned to each possible classifier output. Each element $D(S)_i$ of the vector lies in the range $[0, 1]$ and

$$\sum_{i=1}^{\mathcal{C}+1} D(S)_i = 1. \tag{5.1}$$

Although the vector $D(S)$ has $\mathcal{C}$+1 dimensions, the intrinsic dimensionality of the class label distribution space $\mathcal{L}$ is $\mathcal{C}$. This is due to the fact that equation 5.1 allows one to derive the remaining element in $D(S)$ when $\mathcal{C}$ of the elements are specified.

Since the class label distribution space for the person-people-car classification task is inherently three dimensional, we should be able to visualize this space. To determine the shape of the space in three dimensions, we will investigate the geometric implications of the above constraints on $D(S)$. Consider equation 5.1 once again. If we subtract the fraction of images $D(S)_{reject}$ rejected by the image classifier from both sides of the equation, we obtain

$$D(S)_1 + D(S)_2 + D(S)_3 = 1 - D(S)_{reject}. \tag{5.2}$$

This equation indicates that for a fixed rejection fraction $D(S)_{reject}$, the class label distributions must lie on the plane defined by equation 5.2. Furthermore, since $D(S)_i \in [0, 1]$,

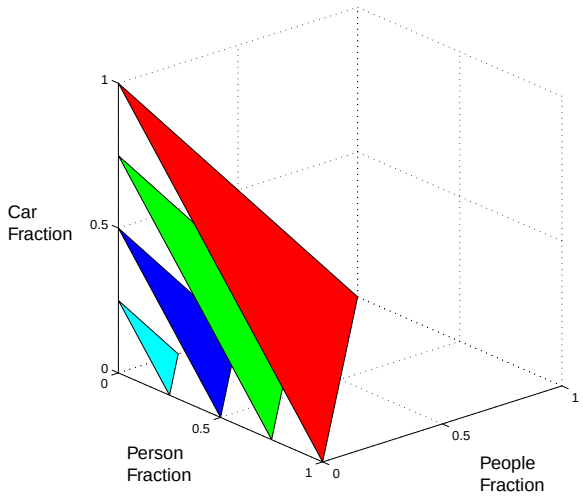


Figure 5.1: Planes of constant rejection fraction

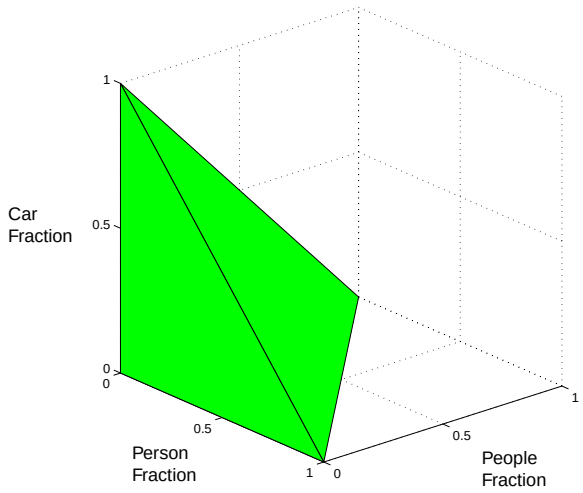


Figure 5.2: Class label distribution space

the class label distributions must also lie within the unit cube in $\mathbb{R}^3$. Therefore only the portion of the plane that is contained within the unit cube is a subset of the class label distribution space $\mathcal{L}$. Figure 5.1 illustrates several planes for various rejection fractions. Since each plane corresponds to the set of all class label distributions with a given rejection fraction, the class label distribution space $\mathcal{L}$ is simply the union of the family of planes

$$D(S)_1 + D(S)_2 + D(S)_3 = 1 - D(S)_{reject} \quad \forall\ D(S)_{reject} \in [0, 1]. \tag{5.3}$$

This leads to a pyramidal volume in $\mathbb{R}^3$ as illustrated in figure 5.2.

5.3 Learning to Classify and Rank the Class Label Distributions

To partition this space, a natural approach is to map the image sequence to the class label that occurs most frequently when classifying the image sequence. Assuming the image

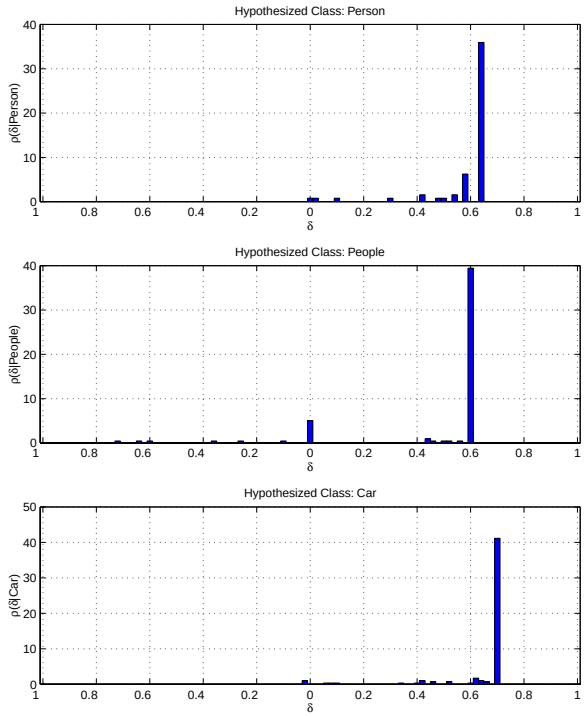


Figure 5.3: Class-conditional discriminant differential densities for the person, people and car classes

classifier produces the correct class label the majority of the time, this simple rule is all that is necessary. In the context of our task, classification is not our only goal. We would like to identify the observed image sequences that produce the lowest levels of classification confidence. Therefore we need some measure of classification confidence that will allow the classification process to rank the image sequences accordingly.

As we have discussed earlier, the approach we will employ involves learning a large differential partition of the class label distribution space. In order to learn this partition and evaluate its performance, we will classify the training and test images using the image classifier and generate the corresponding class label distributions. Once again, we will select a partition from the logistic linear hypothesis class for the class label distribution classifier. Although the logistic linear classifier is unable to form closed decision regions in the class label distribution space, the logistic linear classifier should produce differentials with low likelihoods for novel image sequences that are successfully rejected by the image classifier. Yet a more sophisticated hypothesis class may be required to distinguish between certain types of classifier confusion.

In order to identify the image sequences with the lowest levels of classification confidence, we will rank order the image sequences based upon the maximum likelihood $\rho(\delta_*(D)|\omega_*)$. If the maximum likelihood $\rho(\delta_*(D)|\omega_*)$ increases monotonically with respect to δ_* for all possible class labels ω_*, we will rank order each set of image sequences labeled with a given class label ω_* based on the discriminant differential. Figure 5.3 presents the class-conditional densities $\rho(\delta_k(D)|\omega_k)$ for the various object classes. Given the limited number of test image sequences, it is impossible to state with confidence whether or not the class-conditional densities $\rho(\delta_k(D)|\omega_k)$ monotonically increase with increasing differential δ_k. For simplicity, we will assume the monotonicity does hold.

	Person	*People*	*Car*
Person	**63**	1	0
People	4	**103**	2
Car	0	3	**144**

Table 5.1: Test image sequence confusion matrix

Class	Holdout Estimate	95% Confidence Interval
Person	0.016	[0.000,0.185]
People	0.055	[0.000,0.185]
Car	0.020	[0.000,0.132]

Table 5.2: Class-conditional sequence error rate estimates

Tables 5.1 and 5.2 detail the performance of the resulting logistic linear classifier on the class label distributions for the test image sequences. Clearly, the classifier performs well on the test set. Unfortunately, the limited number of test image sequences leads to loose confidence bounds once again. A larger test set is required to improve our confidence in the generalization performance of the classification process.

In order to visualize the partition produced by the learning procedure, we examined the contours of constant discriminant differential in the plane of constant rejection fraction for $D(S)_{reject} = 0$. This plane corresponds to the plane farthest from the origin in figure 5.1. The contours are shown in figure 5.4. Although there is clearly some bias visible in the contours, the partition that we have learned appears to be approximately equivalent to a simple discriminative rule. Following in the spirit of differential learning, consider ranking the image sequences based on a *class label distribution differential* δ_{CL} defined as

$$\delta_{CL} = \max_{i \in \{1,2,3\}} D(S)_i - \max_{i \neq k, k \in \{1,2,3\}} D(S)_k. \tag{5.4}$$

The contours of constant class label distribution differential in the plane are shown in figure 5.5 for comparison. Since the majority of the training image sequences are classified consistently, it is not surprising to discover that the learned partition is not significantly biased towards certain types of classifier confusion.

To understand what types of classifier confusion are occurring, we examined the scatter of the training and test data in the class label distribution space. Figures 5.6 through 5.8 present density plots of the training and test class label distributions for the various classes along with the contours of constant discriminant differential.[1] Ideally, we hope that the image classifier reliably and consistently classifies the known object image sequences, causing the examples to cluster near the lower right and upper left corners of the density plots. For 80% of the image sequences from each class, the image classifier correctly labels the images consistently. Based on the density plots, we observe that the majority of the remaining image sequences induce one of two types of classifier confusion. Typically when confusion occurs, the image sequences contain a mixture of correctly classified and rejected images. In a number of other examples, the image sequences are assigned a mixture of labels corresponding to the correct class and another related class.

[1] Recall that the classifier cannot discriminate between examples that lie on a contour of constant discriminant differential.

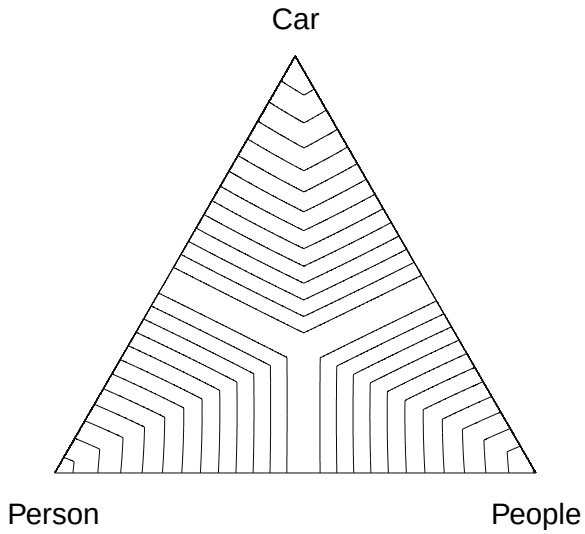


Figure 5.4: Contours of constant discriminant differential for the logistic linear classifier in the plane of constant rejection fraction for $D(S)_{reject} = 0$ (The discriminant differential is maximized at the vertices of the triangle)

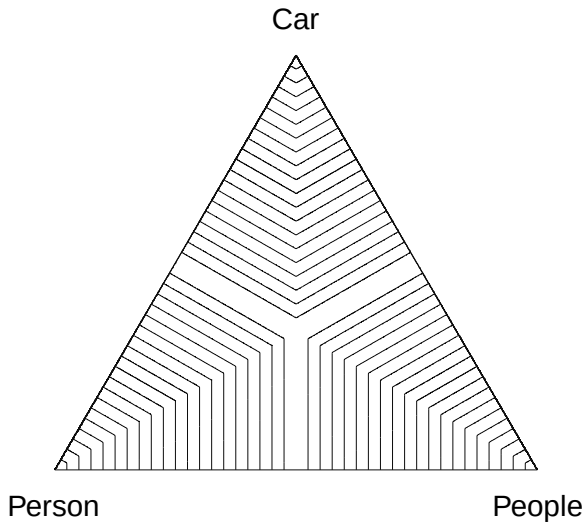


Figure 5.5: Contours of constant class label distribution differential in the plane of constant rejection fraction for $D(S)_{reject} = 0$ (The class label distribution differential is maximized at the vertices of the triangle)

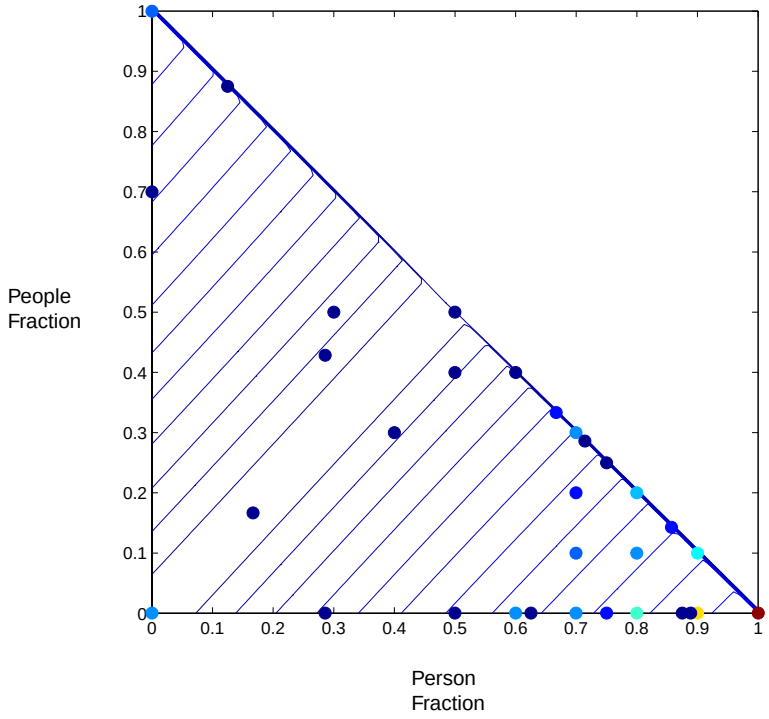


Figure 5.6: Density plot of the class label distribution examples for the person class

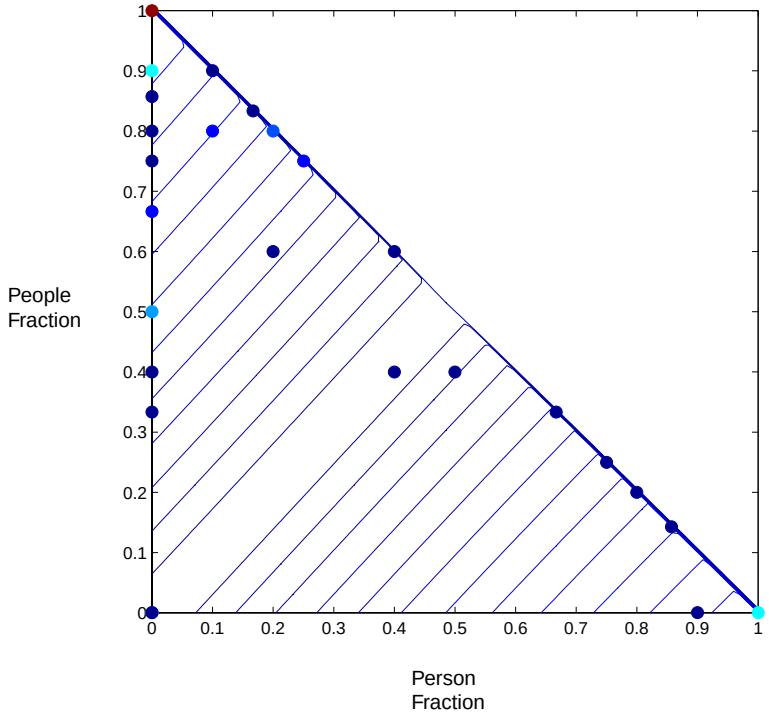


Figure 5.7: Density plot of the class label distribution examples for the people class

	Bicycle	*Truck*	*Van*
Images	148	1409	758
Sequences	25	166	89

Table 5.3: Composition of the dataset for the unknown object classes

5.4 Novel Image Sequence Detection

Now that we have defined the class label distribution classifier and evaluated the response of the classification process when presented with image sequences from the known object classes, we will investigate whether the classification process supports efficient novel image sequence detection. For this evaluation, we assembled another set of image sequences of bicycles, trucks and vans. The composition of the dataset is detailed in table 5.3. These examples were combined with the test image sequences of individuals, groups of people and cars to construct a representative set of observations from a new environment. Our objective is to investigate whether the classification process is able to effectively separate the unknown and known object image sequences by sorting based on the discriminant differentials produced by the class label distribution classifier.

We begin by examining the scatter of the known and unknown object class label distributions in figures 5.9 through 5.11. Except for one bicycle image sequence, all of the unknown object image sequences induce class label distributions that lie in the people-car plane. This implies that the images of the unknown objects are generally rejected or classified as people or car. Considering the scatter plot for the bicycle examples, we find that the majority of the bicycle image sequences induce classifier confusion. Typically, the bicycle image sequences contain images that are classified as people or rejected. Since the majority of the bicycle examples induce more significant classifier confusion than the known object examples labeled as people or car, these examples will be among the first examined by the user, as one would hope.

The scatter plots for the truck and van examples suggest that the classification process is generally unable to distinguish between the various vehicle types. Except for a small number of truck examples that are clustered near the decision boundary, the truck and van image sequences are classified fairly consistently as cars. This is not surprising given the low resolution imagery and limited classifier complexity. Several example image sequences of cars, trucks and vans that are classified consistently as cars are shown in figure 5.12. Examining the binary images, we see that there are subtle differences in the shapes that may support discrimination between the vehicle classes. Yet the image classifier is unable to capitalize on this information.

In order to quantify the novelty detection performance rigorously, two receiver operating characteristic (ROC) curves were generated by varying differential rejection thresholds for the people and car classes. These curves are shown in figures 5.13 and 5.14. Detection performance is often characterized by the area under the ROC curve. This quantity is referred to as the *power of the detector*. If the power is nearly one, the ROC curve must rise rapidly toward the upper left corner of the plot. This indicates that a low error rate and low false alarm rate can simultaneously be achieved. This can only occur if the differential distributions for the known and unknown classes do not overlap significantly. Worst case performance is indicated by a diagonal line with power of 0.5. This scenario is caused by complete overlap of the distributions.

The ROC curves reinforce the conclusions drawn from the scatter plots. Within the set of image sequences labeled as people, the known and unknown object image sequences can be separated fairly well. Approximately 80% of the people image sequences can be

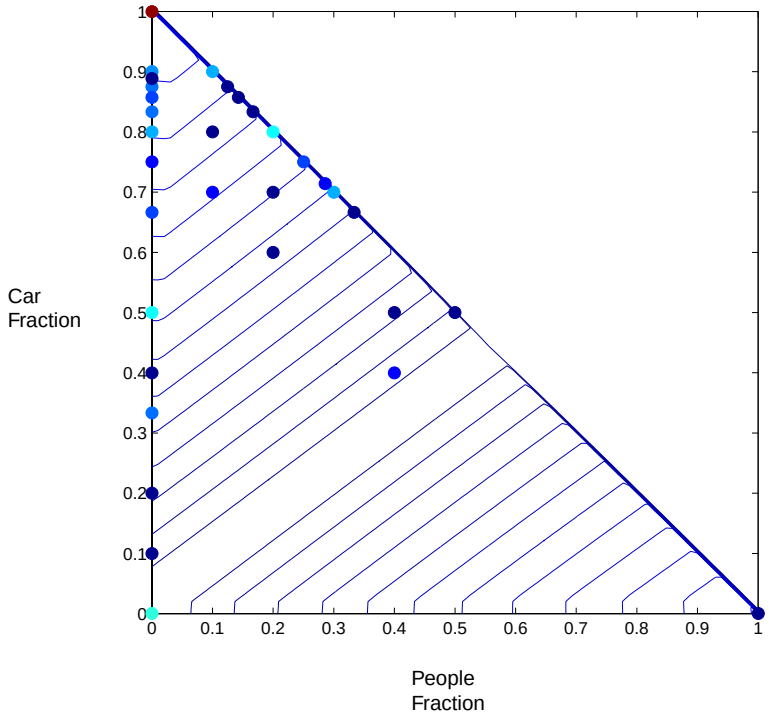


Figure 5.8: Density plot of the class label distribution examples for the car class

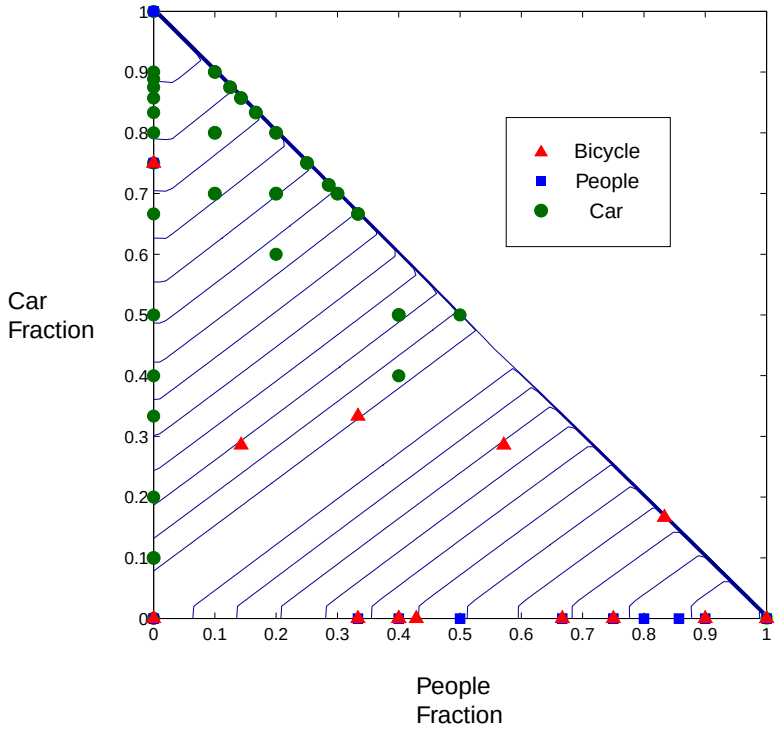


Figure 5.9: Scatter plot of class label distribution examples for the bicycle, people and car classes

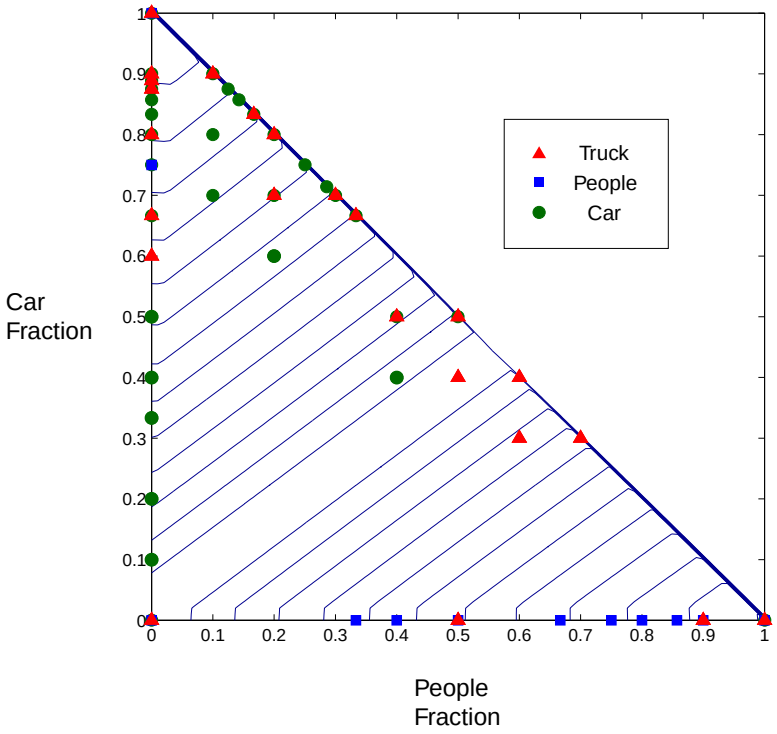


Figure 5.10: Scatter plot of class label distribution examples for the truck, people and car classes

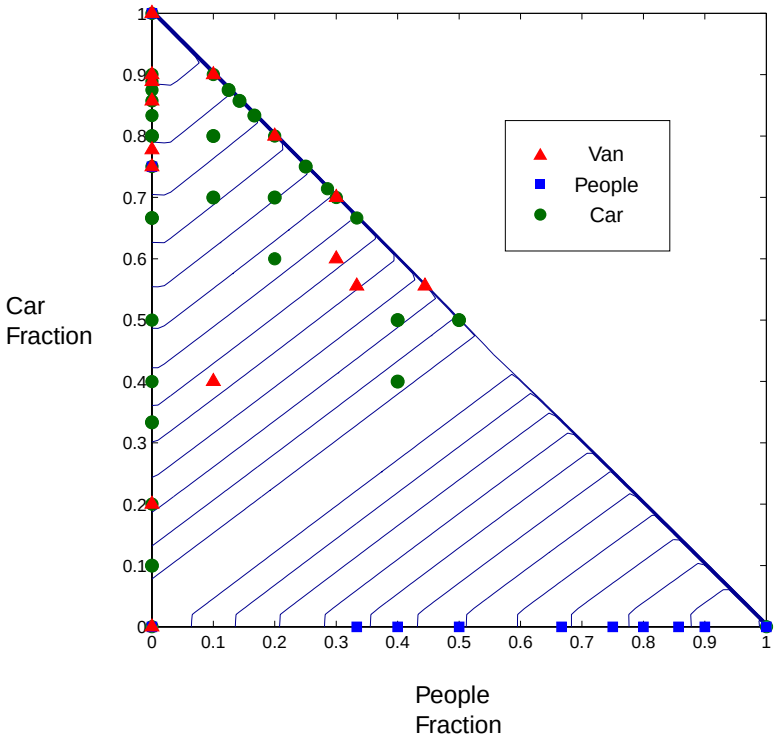


Figure 5.11: Scatter plot of class label distribution examples for the van, people and car classes

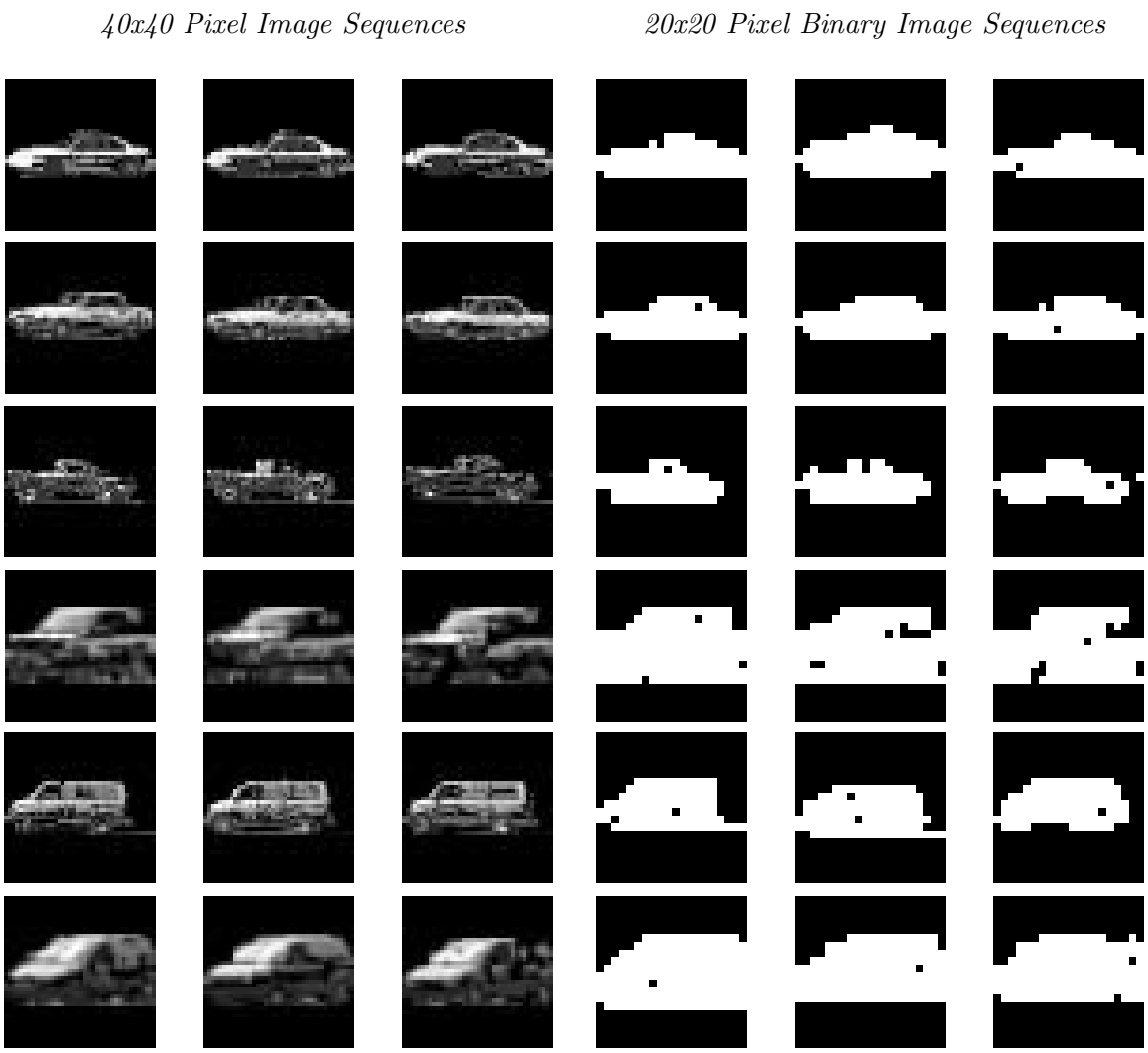


Figure 5.12: Test image sequences of cars, trucks and vans classified consistently as cars

correctly classified while 70% of the bicycle image sequences are rejected. Unfortunately the performance on the vehicle examples is not nearly as encouraging. The ROC curve is essentially a diagonal line indicating the distributions overlap completely.

5.5 Conclusions

Although the classification process delivers promising image classification performance, the utility of the process for efficient novel image sequence detection appears limited in the context of the bicycle-truck-van experiment. Using the logistic linear classifier, the majority of the truck and van image sequences were classified consistently as cars. This causes the majority of the novel vehicle examples to be grouped with the majority of the car examples. Without the aid of additional processes, the user must search through a large fraction of data to identify these examples. This is exactly the outcome we wish to avoid.

The question we need to address now is what changes are necessary to prevent the classification process from placing this burden of interpretation on the user. Since limited classifier complexity leads to a poor representation of the support of the class-conditional

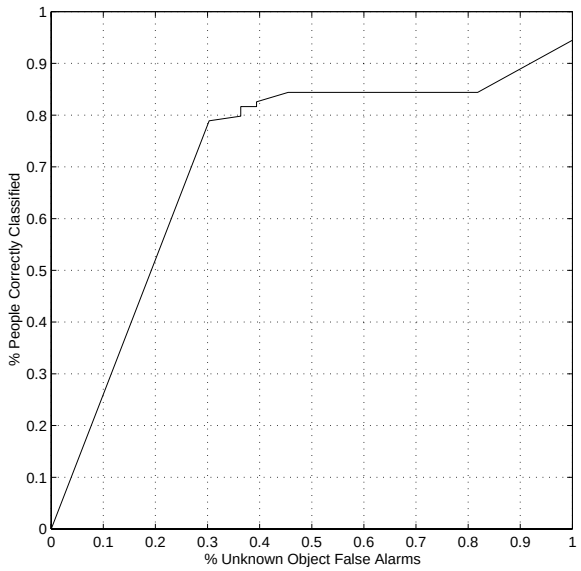


Figure 5.13: Receiver operating characteristic curve generated by varying the differential rejection threshold for the people class

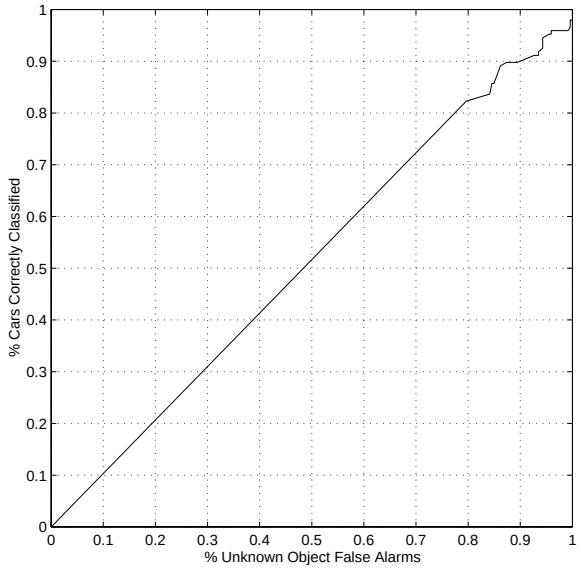


Figure 5.14: Receiver operating characteristic curve generated by varying the differential rejection threshold for the car class

image densities, one option is to employ a more complex hypothesis class. By improving the representation of the support for the known object classes, the image sequences corresponding to the unknown object classes will be more likely to induce classifier confusion.

The problem with this approach is that it is impossible to predict the performance gains obtained at the expense of increased computational complexity. One of our fundamental assumptions is that our knowledge of the environment is incomplete. Therefore the decision regions learned for the known object classes may encompass other unknown object classes. Given our need to minimize the complexity of the classification process to achieve real-time performance, we should focus on designing an efficient classification process for the known object classes that supports novel image sequence detection. Yet we should not expect the classification process to be infallible.

We have been working toward developing an efficient classification process that classifies high dimensional representations of moving objects. By operating in high dimensional spaces, we hope to leverage the available discriminant information to support novel image sequence detection. With the classification process based on the logistic linear image classifier, we have demonstrated the capability to detect novel image sequences that are distinct from the known object classes through classifier confusion. Now we require an additional process to efficiently mine the examples that are classified consistently for additional novel image sequences.

One possible approach involves using one of a myriad of clustering techniques along with a model selection technique to automatically decompose the set of image sequences into a set of image clusters. Assuming the image representation supports discrimination between the known and unknown object classes, we expect that one or more of the image clusters will correspond to the unknown object classes. Therefore the user can examine several images from each cluster to efficiently assess the nature of the dataset.

Within the context of the bicycle-truck-van experiment, it is not clear that such a tool would improve the efficiency of the search dramatically. The low image resolution may not support reliable discrimination between the vehicle classes. Therefore distinct image clusters corresponding to the novel object classes may not exist. In such cases, only a change in the image representation will improve performance. Yet for scenarios where the deficiency rests in the partition, coupling the classification process with an effective image data mining process should provide a foundation for efficient novel image sequence detection.

Chapter 6

Active Incremental Learning

6.1 Overview

In the previous chapter, we evaluated the utility of rank ordering the observed image sequences based on the discriminant differentials produced by the class label distribution classifier. By rank ordering based on the differential, we focus the attention of the user on the image sequences that produce the highest levels of classifier confusion in order to support efficient novel image sequence detection. In this chapter, we will consider whether this sequence selection method also supports efficient incremental learning of the object classes.

6.2 The Process of Incremental Learning

In order to place the problem we are addressing into context, let us begin by presenting our view of the incremental learning process. In contrast to *batch learning* where all of the training data is presented to the learning algorithm at once, *incremental learning* involves training on a series of examples that are presented at different instances in time. Within the surveillance domain, we expect that the learning procedure will be provided with sets of examples. This is in contrast to *online learning* where examples are presented to the learning procedure one at a time.

As the sets of examples are presented to the learning procedure, the classifier must adapt to incorporate the knowledge captured by the new examples. In order to achieve this, modifications may be required to both the hypothesis class and the image representation. These are significant research issues that we will not address in this thesis. Our focus will be on example selection strategies that support rapid acquisition of the underlying concepts from a limited number of labeled examples.

To evaluate the performance of a given example selection strategy, we will conduct a series of experiments where a logistic linear classifier is incrementally trained to classify individuals, groups of people and cars. Over the course of several cycles, small sets of images are selected from unlabeled observations and labeled. Then the current classifier is trained using the available labeled data. The initial classifiers used in all of the experiments result from training on randomly selected sequences from each class. Thirty learning trials are conducted during each experiment beginning with different initial classifiers. During each cycle, the resulting classifier is tested on the disjoint test set used to evaluate the binary image classifier. When comparing the example selection strategies, our interest will lie in the rate of convergence of the sequence image error rate bound.

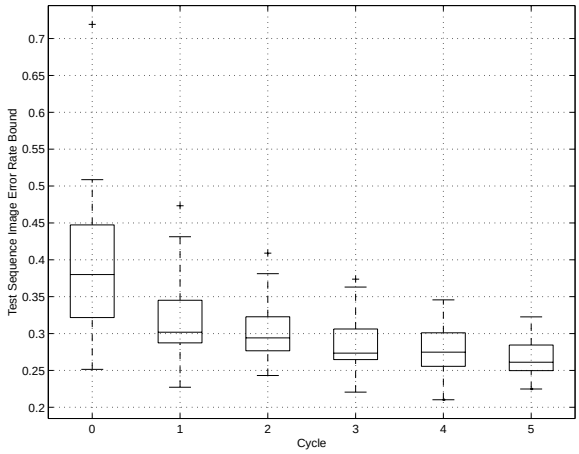


Figure 6.1: Box plots of the test sequence image error rate bounds for random selection when rejection is not permitted

6.3 Example Selection Strategies

6.3.1 Random Image Selection

The first selection strategy we will evaluate is a random selection procedure. This is to establish a baseline for comparison. During each cycle, the unlabeled images are labeled by the current image classifier. Then each image sequence is assigned the class label that occurs most frequently within the sequence. After the image sequences are grouped by class label, images are randomly selected from randomly selected image sequences in each group. Only one image is selected from a given sequence each cycle in order to obtain an independent set of images. This promotes stable convergence of the error rates.

Approximately fifty images are selected from each class initially by randomly selecting sequences. Fifty images are then selected and labeled from each group of classified image sequences in the following cycles. Each learning trial consists of five cycles. This implies approximately 300 images from each class are used to train the image classifier in the final cycle.

Given the limited amount of available training data, we cannot afford to have a separate validation set for parameter adjustment. Learning parameters such as confidence will be adjusted based on k-fold cross-validation in general. In the following experiments, the parameter values are fixed.

Figure 6.1 presents the box plots of the sequence image error rate bounds for each cycle when rejection is not permitted. It appears the error rate bounds converge fairly rapidly using random selection, which suggests the person-people-car classification task is not very challenging. Figure 6.2 compares the box plots of the cycle five error rate bounds with the error rate bounds for the image classifiers trained and tested on the fifty partitions of the dataset. Based on these box plots, we can see that the cycle five classifiers offer comparable performance.

6.3.2 Active Sequence Selection

To improve on the rate of convergence exhibited in figure 6.1, we will now investigate active selection of unlabeled image sequences based on the class label distribution differential defined in equation 5.4. This discriminative rule is used instead of a learned mapping for

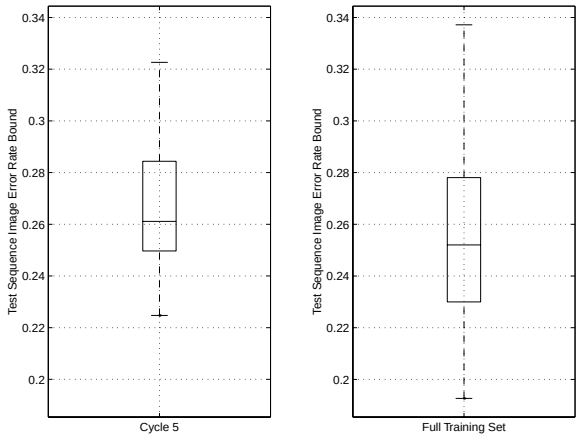


Figure 6.2: (*left*) Box plot of the test sequence image error rate bounds for cycle five of the random selection experiment (*right*) Box plot of the test sequence image error rate bounds for the 50 random partitions of the dataset

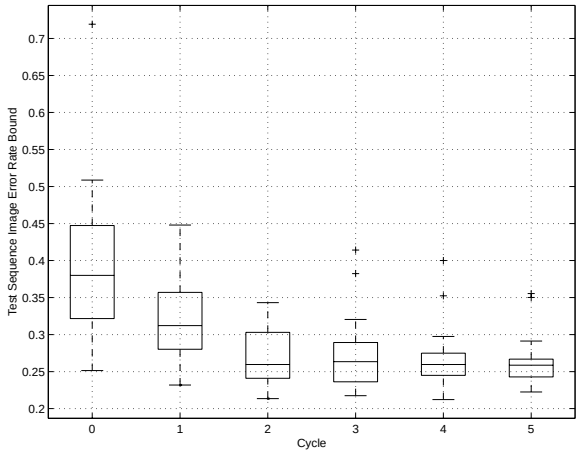


Figure 6.3: Box plots of the test sequence image error rate bounds for active sequence selection when rejection is not permitted

convenience. In this experiment, the image sequences are classified, grouped by class label and sorted based on the differential. Images are then randomly selected from the fifty image sequences producing the lowest differentials in each group. Only one image is selected from a given sequence each cycle.

Figure 6.3 presents the box plots of the sequence image error rate bounds over the five cycles. Figure 6.4 presents the box plots of the reductions in the sequence image error rate bounds achieved with active sequence selection. Based on these figures, we see that active sequence selection does offer an increase in the rate of convergence on average. After two cycles of active selection and incremental training, approximately 75% of the image classifiers offer improvements in performance over their counterparts trained on randomly selected images. Figures 6.5 and 6.6 compare the performance of the cycle two image classifiers from the active sequence selection experiment with the cycle five image classifiers from the random image selection experiment. These figures indicate training the image classifier on the images from the actively selected sequences offers comparable performance

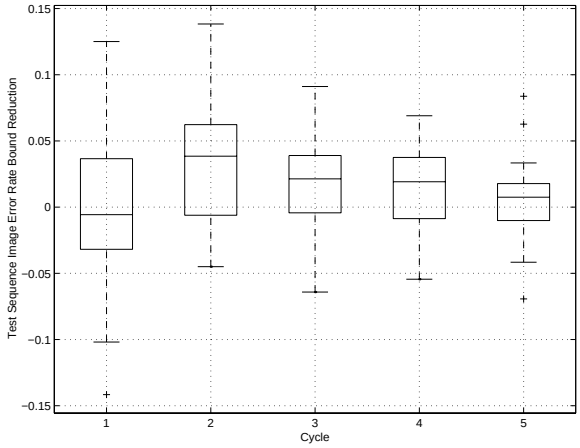


Figure 6.4: Box plots of the reduction in the test sequence image error rate bounds achieved after the transition from random image selection to active sequence selection

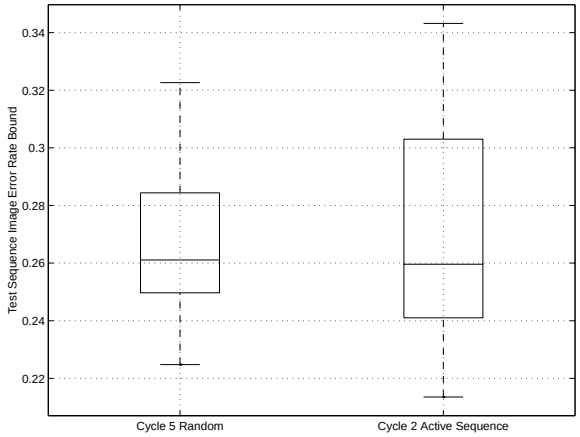


Figure 6.5: Box plots of the test sequence image error rate bounds for cycle 2 of the active sequence selection experiment and cycle 5 of the random image selection experiment

on average with approximately half the number of images.

Although these results suggest active sequence selection offers an advantage over random selection, there are two aspects of this approach that may hinder performance under certain conditions. Our major concern is the effect of randomly sampling from image sequences with significant classifier confusion. If the confusion involves variation in the class labels with minimal rejection, the likelihood of randomly selecting images that are misclassified with large differentials is significant. Such examples may be difficult or impossible to classify correctly. We want to avoid introducing these examples into the training set.

Another weakness may appear if a significant fraction of image sequences are of length one. When an image sequence contains only one image, the image sequence is either classified with maximum or minimum confidence. This implies that if the available unlabeled image sequences are all classified with confidence, the active selection procedure will offer no benefit over random selection. This is due to the fact that the active selection procedure randomly selects an image sequence when more than one image sequence produces the same differential. In order to avoid such potential difficulties, we will examine another technique that actively selects the images directly.

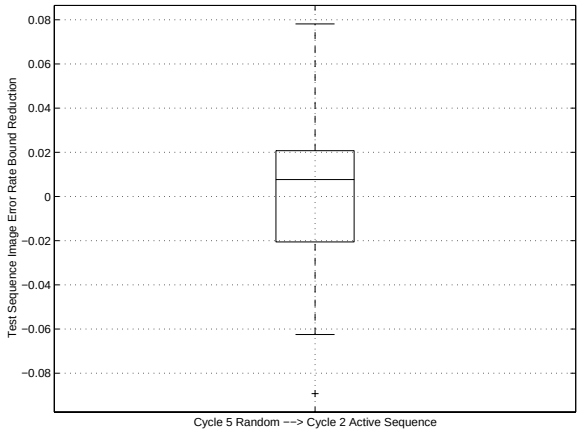


Figure 6.6: Box plots of the reduction in the test sequence image error rate bounds when actively selecting sequences for 2 cycles

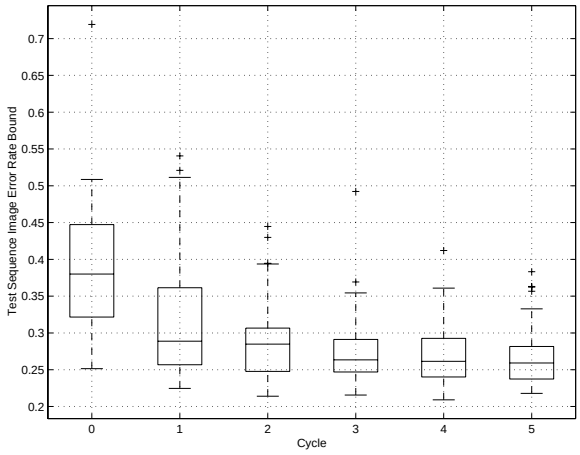


Figure 6.7: Box plots of the test sequence image error rate bounds for active image selection when rejection is not permitted

6.3.3 Active Image Selection

Instead of randomly selecting images from actively selected sequences, we will now investigate active selection of unlabeled images based on the image classifier's discriminant differential. During each cycle, the unlabeled images are classified, grouped by class label and sorted based on the differential. Then the fifty images producing the lowest differentials in each group are selected. The fundamental assumption associated with this selection technique is that the examples closest to the decision boundary are classified with the lowest confidence.

Figure 6.7 presents the box plots of the sequence image error rate bounds over the five cycles. Figure 6.8 presents the box plots of the reductions in the sequence image error rate bounds achieved with active image selection. According to these figures, active image selection does offer some benefit over random selection, but the gains are not particularly significant when compared with those obtained from active sequence selection.

A troubling aspect of figure 6.7 is the fact that the variance of the error rate bound does not appear to decrease significantly after cycle two. After examining the results for several

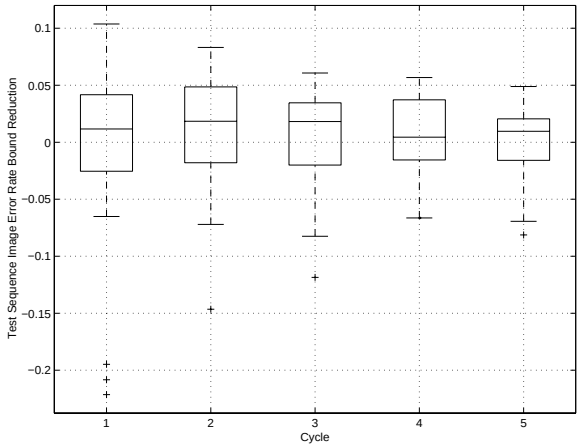


Figure 6.8: Box plots of the reduction in the test sequence image error rate bounds achieved after the transition from random image selection to active image selection

of the learning trials where convergence of the error rate bound is slow, we discovered that the misclassified examples were not necessarily producing smaller differentials than the correctly classified examples on average. More often, the correctly classified and misclassified examples produce differentials of comparable magnitude. In hindsight, this outcome is not surprising. With the ability to translate the images to maximize the differential, there is no compelling reason to believe that the misclassified examples will often lie closer to the decision boundary than correctly classified examples. Therefore we should not expect this strategy to support efficient identification of informative examples when using this image normalization scheme. In light of this discovery, random image selection from the ambiguous sequences is most likely not a detriment to active sequence selection, but the critical component that gives the process an advantage.

6.4 Comparison with Related Example Selection Methods

The two techniques for active example selection that we have investigated in this chapter are closely related to other methods discussed in the literature. Within the last ten years, researchers have focused on methods for actively selecting examples from unlabeled data. The general procedure involves selecting examples that are classified with the highest uncertainty. The discussion within the literature addresses the issue of quantifying the uncertainty.

There appear to be two general approaches to this problem. The first approach involves evaluating the uncertainty by quantifying the variation in the classifications across the set of possible classifiers in the hypothesis class. MacKay [53] estimates the variance of the classifier output in the context of a Bayesian framework. Others [76, 29, 1] evaluate a committee of classifiers and compute a measure of uncertainty based on the classifier outputs. This approach is referred to as *query by committee*.

The second approach discussed in the literature [51, 50, 14] involves selecting unlabeled examples that lie closest to the decision boundary. Although it is widely accepted that this measure provides an inferior assessment of uncertainty relative to the previous approaches, this simple procedure has performed well in a variety of experiments. The most dramatic demonstration of its success was presented by Lewis and Gale [51] in the context of a text classification problem. Using an incremental learning procedure identical to the one

presented in this chapter and a probabilistic classifier, they were able to construct classifiers with comparable performance to classifiers trained on random samples 500 times larger. The improvements obtained by Campbell et al. [14], on the other hand, were not nearly as significant. When actively selecting examples based on the margin to train support vector machines, Campbell et al. obtained sample size reductions of less than an order of magnitude on several problems. As the authors point out, the benefit offered by active example selection is dependent on the complexity of the underlying partition one is attempting to learn.

6.5 Conclusions

In this chapter, we have shown that active sequence selection based on the class label distribution differential supports efficient incremental learning of the object classes in the context of the person-people-car classification task. We have also addressed concerns about the performance of the active sequence selection procedure by comparing the process against active image selection based on the discriminant differential. Although active example selection based on the margin/differential has proven to be successful in other experiments, we have seen that active sequence selection is more appropriate for the specific image classifier used in these experiments. In general, we expect that active sequence selection will prove useful when classifying image sequences in the manner outlined in this thesis. It will be interesting to investigate whether active image selection offers improvements in performance when the image normalization procedure is not employed.

Chapter 7

Implementation

7.1 Overview

Throughout this thesis, we have strived to develop a strategy for designing computationally efficient classification processes that support collaborative classification. In the previous chapters, we have evaluated the performance of the process for the person-people-car task through a variety of experiments. Yet we have not investigated the runtime performance of the process. In this chapter, we present an overview of the CMU Cyberscout distributed video surveillance system, which employs the classification process, and provide performance results for the current system.

7.2 CyberARIES: Agent-Based Software Architecture

7.2.1 Functional Requirements

In order to realize a distributed surveillance system, we must first define a framework for coordinated sensing, processing and communication among the sensing systems. The path that we have chosen to follow toward this objective involves defining an agent-based software architecture for collaboration among individual sensor systems. Ideally, this architecture will allow the operator to provide the distributed surveillance system with high level surveillance objectives which in turn are decomposed automatically into a collection of taskings for individual sensor systems. As the sensor systems collect data about the environment, they will collaborate with one another to assemble a common interpretation of the environment. This will involve collaborative processing and control for detection, classification and tracking. Once an interpretation is formed, the taskings for individual sensor systems are then automatically updated in order to resolve remaining ambiguity or improve the system's ability to assess general activity in the environment. This way, the user achieves maximum information gain with minimal input. In this section, we define the basic components of the CyberARIES architecture designed to support these objectives.[1]

7.2.2 Architecture Fundamentals

7.2.2.1 The Agent

We begin our overview of the CyberARIES architecture by defining the basic building block: the agent. We define an agent as software with the following properties:

[1]The principal architect of the CyberARIES architecture is Mahesh Saptharishi <mahesh@andrew.cmu.edu>.

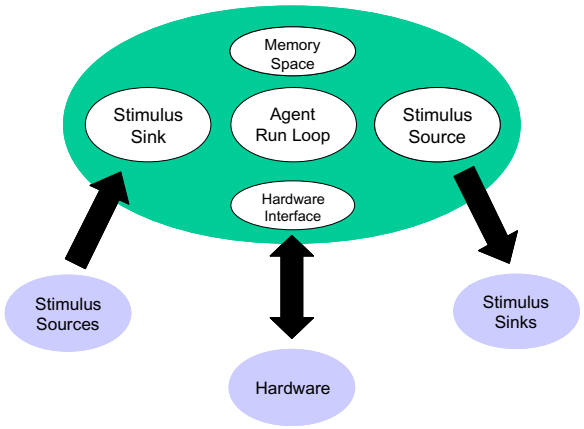


Figure 7.1: General agent structure within CyberARIES

- Accepts stimuli from other agents
- Has steady state behavior in the absence of stimuli
- Can provide stimuli to other agents

The general agent structure within CyberARIES is shown in figure 7.1. The behavior of the agent is defined by the *agent run loop* which is executed until either the operator terminates the agent or the agent terminates itself. Communication between agents occurs through connections between stimulus sources and sinks. A *stimulus sink* receives all incoming messages (stimuli) from other agents and stores them until the agent run loop requests a stimulus. A *stimulus source* receives stimuli from the agent run loop and attempts to transmit them to the stimulus sink of the specified agent(s).

7.2.2.2 The Distribution Layer

Given that there will be no centralized or hierarchical control of the agents in the system, communication among the agents will form the basis for allocation of processing and sensing resources across the distributed surveillance system. Within CyberARIES, the *distribution layer* is the communications infrastructure that is responsible for routing stimuli between agents and regulating the flow of stimuli within the system [2]. When an agent wishes to send a stimulus to another agent, the distribution layer handles the details of establishing the necessary connections to deliver the stimulus. During transmission, it also monitors the arrival rate of stimuli relative to the processing rate of the receiving agents to ensure that agents are not being overloaded. If the distribution layer detects a problem, it will ask transmitting agents to reduce their rate of transmission.

Upon receiving such a request, an agent communicates with other agents to determine if another agent has excess processing capacity to handle additional stimuli. If an agent accepts the processing task, some stimuli bound for the original receiving agent are transmitted to the volunteer. Otherwise, if no agent can accept additional stimuli, the transmitting agent simply sleeps for a certain amount of time during each cycle of the agent run loop in order to reduce its transmission rate. Using such simple interactions among agents, we obtain a means for *dynamic load balancing* which utilizes emergent collaboration among the agents to achieve these ends.

[2] In keeping with the spirit of our distributed architecture, the distribution layer is composed of a set of distribution agents with one running on every processor in the distributed surveillance system.

Figure 7.2: CyberScout ATV

Similar constraint-based processes that utilize the distribution layer may also lead to *emergent collaboration for perception* as well. In such tasks, one challenge is to automatically determine which agents should receive information derived from sensory data by a given agent. By utilizing some utility measure, agents receiving such information can provide feedback to the distribution layer which in turn can be used to allocate communication capacity to those connections which provide the most significant gains in perception performance. In this way, the distributed surveillance system can learn to exchange information among the agents in a manner that most effectively reduces uncertainty in the interpretation of activities in the environment.

7.2.3 CyberScout Agent-Based Framework

In the CyberScout program, we utilize two retrofitted Polaris all-terrain vehicles (ATVs) along with stationary sensor systems to perform tactical surveillance. One of the ATVs is shown in figure 7.2. On each sensor system, one or multiple processors host a set of agents that perform a variety of functions for perception, planning and control. Perception agents are responsible for such tasks as change detection, classification, tracking, obstacle avoidance and landmark detection. In the next section, we will focus on the perception processes for surveillance.

7.3 Perception for Surveillance

7.3.1 Process Collaboration

As we have discussed previously, there are three processes that interpret the video streams: change detection, region tracking and classification. Figure 7.3 illustrates the flow of information between the processes as the video is processed. The change detection process nominates regions of significant intensity change in a given video frame and passes these regions to the classification and region tracking processes. The classification process classifies the regions and passes the class labels to the region tracking process. The region tracking process attempts to match the candidate regions with regions from the previous frame and passes the associations to the classification process. The classification process then updates the object classifications based upon the class label histories of the objects. In the next

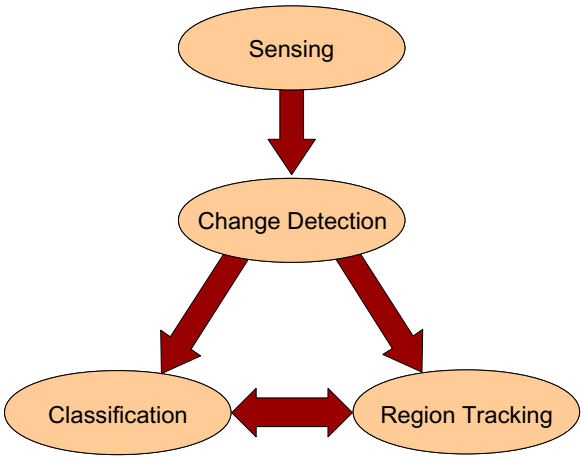


Figure 7.3: Collaboration among the perception processes

section, we briefly review the change detection and region tracking processes employed in the CyberScout system.

7.3.2 Process Descriptions

7.3.2.1 Change Detection

Typical background subtraction algorithms estimate the dominant mode of the video using an autoregressive (AR) filter and nominate regions of the current frame where significant intensity differences exist between the current frame and the background model. Such processes are computationally efficient but are also sensitive to camera jitter and moving foliage. The change detection process we employ [68] overcomes the weaknesses of the standard background subtraction algorithm by supporting multi-model representations of the background. Instead of capturing the dominant mode with a single AR filter, a set of AR filters is used to estimate the centers and widths of the modes of the intensity distribution. When a pixel value lies within the range of one mode, the pixel is assigned to the background and the background model is updated. Otherwise, the pixel is nominated as a motion pixel. Typically no more than four modes are required for a robust background model.

7.3.2.2 Region Tracking

In order to robustly correspond regions across a series of frames, the region tracking process [69] considers the position and appearance variation of the regions. Using linear prediction with multiple hypothesis tracking, the process attempts to correspond regions based on the predicted positions of the tracked regions. If ambiguity arises, the candidate regions are compared with the reference region from the previous frame to determine the best match. The mechanism for matching regions is based on linear classifiers that are trained offline to associate regions from a given class. An online procedure then adapts the weights of the baseline linear classifiers for each tracked object to emphasize distinct features that simplify the discrimination task.

7.3.3 Performance

On both the mobile and stationary sensor systems, one processor is dedicated to each video stream. Within the current system, there are a mixture of Pentium II 350 MHz and Pentium

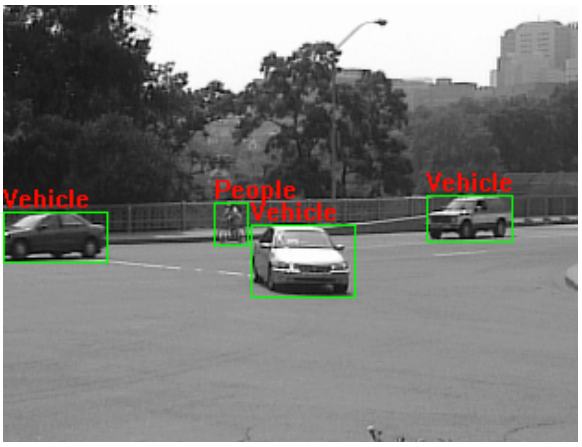


Figure 7.4: Classifications overlaid on the original video

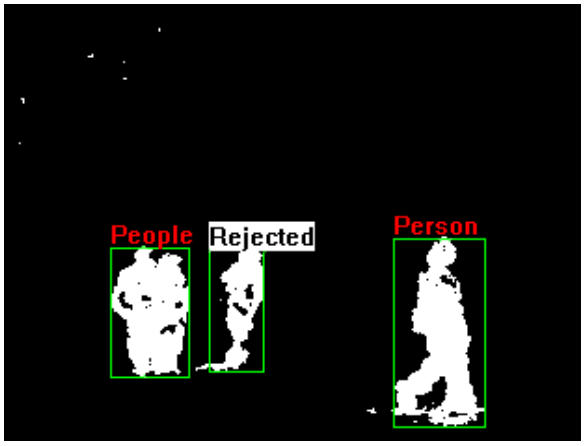


Figure 7.5: Classifications overlaid on the binary motion image

III 500 MHz platforms with 128 Mb of RAM. Depending on the available computational power and the number of objects in the scene, a given sensor system will generally operate between 5 and 10 Hz. Figures 7.4 and 7.5 present classification results from the system while in operation. During our evaluations of the system, we noticed that the image classifier was having difficulty discriminating between the *person* and *people* classes when pairs of people walked in close proximity such that one person partially occluded the other. This problem was alleviated by processing all horizontal translations of the regions up to 10 pixels in either direction.

7.4 Conclusions

The current system provides excellent performance at a rate that supports the timely assessment of the state of the environment. Unfortunately, with the current classification process, there is little computational margin left to increase the complexity of the classification process. As we discussed earlier in the thesis, we must employ a hierarchical classification scheme and avoid the current image normalization procedure to address more complex classification tasks. Since 63 dot products are computed for each image currently, there should be enough computational resources to address more complex tasks using a hierarchical approach. Yet we will need to investigate whether image normalization based on the center of mass will yield adequate rejection performance.

Chapter 8

Conclusions

8.1 Thesis Review

A distributed video surveillance system produces volumes of data when observing a given environment. In order to derive relevant information from the data in a timely fashion, tools are needed to limit the burden of interpretation on the user. Ideally, we would like to maximize the amount of relevant information provided to the user while minimizing the amount of context required from the user. In the introduction, we presented the interpretation cycle as a candidate process for achieving this objective. This process relies on collaboration between the system and the user to incrementally acquire the context necessary to interpret the environment. Fundamentally, the success of the process depends on the ability of the system to efficiently classify image sequences and identify novel image sequences for efficient incremental learning of the object classes.

As we have discovered in this thesis, simultaneously satisfying these objectives is non-trivial. In order to achieve computational efficiency, we need to employ a low complexity image classifier and rely on the ability to observe moving objects over time from a variety of perspectives to achieve robust performance. Yet in order to support novel image sequence detection, we need to partition a high dimensional feature space. This may require additional complexity to successfully detect unknown objects and novel views of known objects. The degree of our success will be determined by the complementary nature of the image representation and hypothesis class.

To learn a partition of a given high dimensional feature space, we explored the utility of large margin classification techniques. Theoretical and experimental results clearly support the effectiveness of these techniques. Yet the common approaches of support vector learning and boosting do not yield efficient representations of the partition due to the nature of their construction. Differential learning provides another option to avoid this pitfall. Since the hypothesis class must be specified prior to learning, we can constrain our search for efficient large differential partitions to low complexity hypothesis classes.

Once the partition is learned, the definition of the image classifier is not complete. We must constrain the decision regions by estimating the discriminant differential rejection thresholds. Ideally, we would like to constrain the decision regions to regions of feature space where there is sufficient data to support the decision. Unfortunately, a low complexity image classifier that successfully classifies the known object classes may yield a poor representation of the support of the class-conditional image densities for the known object classes. In such cases, modifications to the image representation or hypothesis class may improve the resulting partition. Yet the complexity of the image classifier is ultimately constrained by the available computation.

Due to the limitations of the image representation and hypothesis class, we should not

expect novel image sequences to always induce classifier confusion. In cases where a given object presents views that are distinctly different from those encountered in the training set, there is a high likelihood of observing classifier confusion. In other scenarios where the deviations are less significant, classifier confusion is not as likely to occur. Therefore we require an image data mining process to search for novel image sequences that are classified consistently. We believe the combination of these methods for novelty detection will support efficient incremental learning of object classes.

8.2 Contributions

Within the video surveillance domain, real-time surveillance systems have been developed that interpret the environment based on context specified prior to deployment. Yet no process has been introduced that facilitates efficient incremental learning of context. Our main contribution in this thesis is the definition of a general classification process and associated principles for classifier design that support real-time image sequence classification, novel image sequence detection and incremental learning. Specific additional contributions include:

- Generalization of the conditions on the classification figure-of-merit (CFM) objective function that must be satisfied in order to induce large differential partitions.
- Definition of a bound on the worst case performance of the image classifier based on the sequence image error rate.
- Definition of minimax differential learning for learning minimax partitions of feature space.
- Demonstration of a real-time appearance-based process for person-people-vehicle classification and confidence assessment.

8.3 Future Work

During the course of our investigation, we addressed a series of issues relating to the design and implementation of the classification process. A number of avenues remain to be explored. Interesting directions for future research include:

- Modifying the learning procedure to directly estimate the support of the class-conditional densities.
- Investigating procedures for adapting the image representation and hypothesis class to support incremental learning.
- Defining a procedure for learning computationally efficient, hierarchical large differential classifiers.
- Developing methods for mining the observations for novel image sequences.

Appendix A

Bounding a Classifier's Error Rate

In order to evaluate generalization performance, a fraction of the available data is generally withheld from the classifier design process and used to obtain an unbiased estimate of the classifier's error rate. Often such estimates are based upon a limited test set. Therefore we would like to quantify our uncertainty in this estimate by computing associated confidence bounds.

Given a disjoint, labelled test set

$$\mathcal{T}_M = \left\{ \left(X^{T_1}, \omega^{T_1} \right), \left(X^{T_2}, \omega^{T_2} \right), \ldots, \left(X^{T_M}, \omega^{T_M} \right) \right\}, \tag{A.1}$$

resulting from a series of independent trials, the *holdout estimate* $\hat{\mathrm{P}}_e$ of the error rate for the classifier $\mathcal{D}(X)$ is defined as

$$\hat{\mathrm{P}}_e = \frac{1}{M} \sum_{i=1}^{M} e_{T_i} \tag{A.2}$$

where

$$e_{T_i} = \mathcal{I}_{\left\{ \mathcal{D}(X^{T_i}) \neq \omega^{T_i} \right\}}. \tag{A.3}$$

In this section, we will introduce an upper bound on the probability

$$\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N \right) \tag{A.4}$$

that the actual probability of error

$$\mathrm{P}_e = \mathrm{E}\left[\mathcal{D}(X) \neq \omega \middle| \mathcal{S}^N \right] \tag{A.5}$$

exceeds the holdout estimate $\hat{\mathrm{P}}_e$ by a margin ϵ given the training set $\mathcal{S}^N$. This bound is a special case of *Hoeffding's inequality* [40] which provides an upper bound on the probability that the sum of a finite collection of bounded, independent random variables exceeds its mean by a given margin. Our general overview of the derivation utilizes elements from the proofs presented in [40] and [22].

The approach employed for bounding the probability relies on the following observation. The probability in (A.4) can be reexpressed as the expectation

$$\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N \right) = \mathrm{E}\left[\mathcal{I}_{\{\mathrm{P}_e - \hat{\mathrm{P}}_e - \epsilon \geq 0\}} \middle| \mathcal{S}^N \right]. \tag{A.6}$$

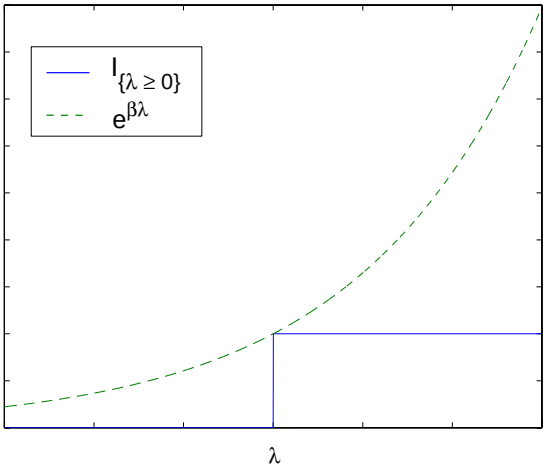


Figure A.1: Bounding the indicator function

Since

$$\mathcal{I}_{\{\lambda \geq 0\}} \leq e^{\beta\lambda} \tag{A.7}$$

for $\beta \geq 0$ as depicted in figure A.1,

$$\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N\right) \leq \mathrm{E}\left[e^{\beta(\mathrm{P}_e - \hat{\mathrm{P}}_e - \epsilon)} \middle| \mathcal{S}^N\right]. \tag{A.8}$$

The remaining task is to evaluate the expectation and minimize the bound with respect to β. This approach is known as the *Chernoff bounding method* [15].

In order to simplify matters, we will begin by restating the probability we wish to bound as

$$\begin{aligned}
\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N\right) &= \mathrm{P}\left(\mathrm{P}_e - \frac{1}{M}\sum_{i=1}^{M} e_{T_i} \geq \epsilon \middle| \mathcal{S}^N\right) && \text{(A.9)} \\
&= \mathrm{P}\left(\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i} \geq M\epsilon \middle| \mathcal{S}^N\right). && \text{(A.10)}
\end{aligned}$$

Applying the above observation and manipulating the right hand side, we find

$$\begin{aligned}
\mathrm{P}\left(\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i} \geq M\epsilon \middle| \mathcal{S}^N\right) &= \mathrm{E}\left[\mathcal{I}_{\{\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i} - M\epsilon \geq 0\}} \middle| \mathcal{S}^N\right] && \text{(A.11)} \\
&\leq \mathrm{E}\left[e^{\beta\left(\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i} - M\epsilon\right)} \middle| \mathcal{S}^N\right] && \text{(A.12)} \\
&= e^{-\beta M\epsilon}\, \mathrm{E}\left[e^{\beta\left(\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i}\right)} \middle| \mathcal{S}^N\right] && \text{(A.13)} \\
&= e^{-\beta M\epsilon} \prod_{i=1}^{M} \mathrm{E}\left[e^{\beta(\mathrm{P}_e - e_{T_i})} \middle| \mathcal{S}^N\right]. && \text{(A.14)}
\end{aligned}$$

To bound the final expectation on the right hand side, we employ the bound

$$\mathrm{E}\left[e^{sX}\right] \leq e^{\frac{s^2(b-a)^2}{8}} \tag{A.15}$$

where $\mathrm{E}X = 0$, $a \leq X \leq b$ and $s > 0$. A proof of this bound is presented in [22]. Given $\mathrm{E}\left[\mathrm{P}_e - e_{T_i} \middle| \mathcal{S}^N\right] = 0$ and $\mathrm{P}_e - 1 \leq \mathrm{P}_e - e_{T_i} \leq \mathrm{P}_e$,

$$\mathrm{E}\left[e^{\beta(\mathrm{P}_e - e_{T_i})} \middle| \mathcal{S}^N\right] \leq e^{\frac{\beta^2}{8}}. \tag{A.16}$$

Utilizing this result, we obtain

$$\begin{aligned} \mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N\right) &= \mathrm{P}\left(\sum_{i=1}^{M} \mathrm{P}_e - e_{T_i} \geq M\epsilon \middle| \mathcal{S}^N\right) && \text{(A.17)} \\ &\leq e^{-\beta M\epsilon} \prod_{i=1}^{M} e^{\frac{\beta^2}{8}} && \text{(A.18)} \\ &= e^{\frac{\beta^2 M}{8} - \beta M\epsilon}. && \text{(A.19)} \end{aligned}$$

Minimizing the bound with respect to β, we find $\beta = 4\epsilon$ which produces the bound

$$\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N\right) \leq e^{-2M\epsilon^2}. \tag{A.20}$$

Following a similar argument, one can also prove that

$$\mathrm{P}\left(\hat{\mathrm{P}}_e - \mathrm{P}_e \geq \epsilon \middle| \mathcal{S}^N\right) \leq e^{-2M\epsilon^2} \tag{A.21}$$

[22]. Therefore if one combines the previous two bounds, another useful bound

$$\mathrm{P}\left(|\hat{\mathrm{P}}_e - \mathrm{P}_e| \geq \epsilon \middle| \mathcal{S}^N\right) \leq 2e^{-2M\epsilon^2} \tag{A.22}$$

is obtained.

Given the convenient form of the resulting bounds, the computation of various types of confidence bounds is trivial. For example, if we require an upper bound on the holdout estimate $\hat{\mathrm{P}}_e$ such that

$$\mathrm{P}\left(\mathrm{P}_e - \hat{\mathrm{P}}_e \geq \epsilon \middle| \mathcal{S}^N\right) \leq 1 - \alpha, \tag{A.23}$$

we can solve the equation

$$e^{-2M\epsilon^2} = 1 - \alpha \tag{A.24}$$

to obtain the maximum deviation

$$\epsilon = \sqrt{\frac{-1}{2M} \ln(1 - \alpha)}. \tag{A.25}$$

Appendix B

Minimax Differential Learning

B.1 Introduction

The standard assumption often invoked in statistical pattern classification problems is that the process giving rise to the training sample is stationary. This implies that the training sample is representative of the types of examples we shall see in the future which gives us hope of designing a classifier that will generalize to unseen examples. Unfortunately in the context of surveillance, the class priors for a given environment are often a function of time. In addition, class priors can vary significantly across environments. This leads us to question minimizing the error rate over the training sample when the class priors are unknown. We propose instead to minimize the maximum class-conditional error rate, thereby minimizing the worst case error rate. We will investigate whether this is feasible in the context of differential learning using the synthetic CFM objective function.

B.2 The Minimax Condition

To begin, let us first consider the general problem of minimizing the maximum class-conditional error rate. Given a classifier that partitions the feature space $\mathbf{X}$ into $\mathcal{C}$ decision regions $\{R_1, R_2, \ldots, R_{\mathcal{C}}\}$, the classifier's probability of error can be expressed as

$$\mathrm{P_e} = \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_i)\mathrm{P}_{\mathrm{e}|\omega_i} \tag{B.1}$$

$$= \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_i) \int_{\overline{R}_i} \rho(\mathrm{X}|\omega_i)\mathrm{dX}. \tag{B.2}$$

where $\overline{R}_i = \mathbf{X} - R_i$. Assuming the class priors are unknown, the probability of error can vary over the range defined by the minimum and maximum class-conditional error rates. In the malicious environment, only examples from the class with the maximum class-conditional error rate are presented to the classifier. Our goal is to adjust the decision regions such that the worst case performance is minimized.

For simplicity, we will discuss the two class problem first. Assume for the moment that the class-conditional error rates for the two classes are not equal. Using the notation $\mathrm{P}_{\mathrm{e}|\omega_{(i)}}$

to indicate the *ith* largest class-conditional error rate, our assumption implies

$$\begin{aligned} \mathrm{P}_{\mathrm{e}|\omega_{(1)}} &> \mathrm{P}_{\mathrm{e}|\omega_{(2)}} && \text{(B.3)} \\ \int_{\overline{R}_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} &> \int_{\overline{R}_{(2)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} && \text{(B.4)} \\ &> 1 - \int_{\overline{R}_{(1)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} && \text{(B.5)} \end{aligned}$$

In order to minimize $\mathrm{P}_{\mathrm{e}|\omega_{(1)}}$, region $R_{(1)}$ must be expanded. We will now demonstrate that as $R_{(1)}$ expands, the error rate difference $\mathrm{P}_{\mathrm{e}|\omega_{(1)}} - \mathrm{P}_{\mathrm{e}|\omega_{(2)}}$ decreases, as one would anticipate.

If the new decision region $R'_{(1)} = R_{(1)} + \Delta R_{(1)}$, the complements of regions $R'_{(1)}$ and $R'_{(2)}$ are

$$\begin{aligned} \overline{R'}_{(1)} &= \mathbf{X} - R'_{(1)} \\ &= \mathbf{X} - R_{(1)} - \Delta R_{(1)} \\ &= \overline{R}_{(1)} - \Delta R_{(1)} && \text{(B.6)} \\ \overline{R'}_{(2)} &= \mathbf{X} - R'_{(2)} \\ &= \mathbf{X} - \overline{R}_{(1)} + \Delta R_{(1)} \\ &= \overline{R}_{(2)} + \Delta R_{(1)}. && \text{(B.7)} \end{aligned}$$

Therefore

$$\begin{aligned} &\int_{\overline{R'}_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} - \int_{\overline{R'}_{(2)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} \\ &= \int_{\overline{R}_{(1)} - \Delta R_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} - \int_{\overline{R}_{(2)} + \Delta R_{(1)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} \\ &= \int_{\overline{R}_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} - \int_{\overline{R}_{(2)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} - \\ &\quad \int_{\Delta R_{(1)}} \rho(\mathrm{X}|\omega_{(1)}) + \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX}. \qquad \text{(B.8)} \end{aligned}$$

Since

$$\int_{\Delta R_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} > 0 \qquad \text{(B.9)}$$

in order to obtain a reduction in the class-conditional error rate $\mathrm{P}_{\mathrm{e}|\omega_{(1)}}$,

$$\int_{\Delta R_{(1)}} \rho(\mathrm{X}|\omega_{(1)}) + \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} > 0. \qquad \text{(B.10)}$$

This implies

$$\begin{aligned} &\int_{\overline{R'}_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} - \int_{\overline{R'}_{(2)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX} \\ &< \int_{\overline{R}_{(1)}} \rho(\mathrm{X}|\omega_{(1)})\mathrm{dX} - \int_{\overline{R}_{(2)}} \rho(\mathrm{X}|\omega_{(2)})\mathrm{dX}. \qquad \text{(B.11)} \end{aligned}$$

So as the partition is iteratively refined, the class-conditional error rates converge. Ultimately the minimization produces a partition where

$$\int_{\overline{R}_1} \rho(\mathrm{X}|\omega_1)\mathrm{dX} = \int_{\overline{R}_2} \rho(\mathrm{X}|\omega_2)\mathrm{dX} = \epsilon. \tag{B.12}$$

Substituting this result into equation B.2, we find that

$$\begin{aligned} \mathrm{P_e} &= \epsilon \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_i) & \text{(B.13)} \\ &= \epsilon & \text{(B.14)} \end{aligned}$$

which clearly indicates the error rate for this classifier is invariant to the class prior probability distribution.

This argument extends to the $\mathcal{C}$ class problem as well. As long as there exists at least one class with an error rate less than the class(es) with the maximum class-conditional error rate, the decision regions can be modified to reduce the maximum class-conditional error rate while increasing the class-conditional error rates of one or more classes. A minimax partition is achieved when the condition

$$\int_{\overline{R}_1} \rho(\mathrm{X}|\omega_1)\mathrm{dX} = \int_{\overline{R}_2} \rho(\mathrm{X}|\omega_2)\mathrm{dX} = \ldots = \int_{\overline{R}_\mathcal{C}} \rho(\mathrm{X}|\omega_\mathcal{C})\mathrm{dX} \tag{B.15}$$

is satisfied for the specified partition. Ideally, we would like to identify the optimal minimax partition which obtains the minimum error rate over the set of all possible minimax partitions. Yet in practice, we will be limited to searching for the minimax partition admitted by the hypothesis class with the minimum error rate, if such a partition exists.

B.3 Minimax Learning and CFM

Now that we understand the condition that must be satisfied in order to obtain a minimax classifier, we will investigate whether a learning strategy similar to the approach above can be used in conjunction with the synthetic classification figure-of-merit (CFM) objective function to learn minimax classifiers. We will assume that an unlimited amount of training data is available. We begin by evaluating CFM once again in the limit of infinite training data. Recall that CFM is expressed as

$$\mathrm{CFM}\left(\mathcal{S}^N|\theta\right) = \frac{1}{N}\sum_{i=1}^{N} \sigma(\delta_k(\mathrm{X}^i|\theta), \psi) \tag{B.16}$$

for a finite sample of size N where the class label for the i^{th} example $\omega^i = \omega_k$. As the number of training examples grows asymptotically large, $\mathrm{CFM}(\mathcal{S}^N|\theta)$ converges in probability to the expected value of CFM over $\mathcal{X} \times \Omega$ where Ω is the class label space. Expanding $\mathrm{E}_{\mathbf{X},\Omega}[\sigma(\delta(\mathrm{X}|\theta), \psi)]$, we obtain

$$\begin{aligned} \mathrm{E}_{\mathbf{X},\Omega}[\sigma(\delta(\mathrm{X}|\theta), \psi)] &= \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_c)\mathrm{E}_{\mathbf{X}|\Omega}[\sigma(\delta(\mathrm{X}|\theta), \psi)|\omega_c] & \text{(B.17)} \\ &= \sum_{c=1}^{\mathcal{C}} \mathrm{P}(\omega_c) \int \sigma(\delta_c(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_c)\mathrm{dX}. & \text{(B.18)} \end{aligned}$$

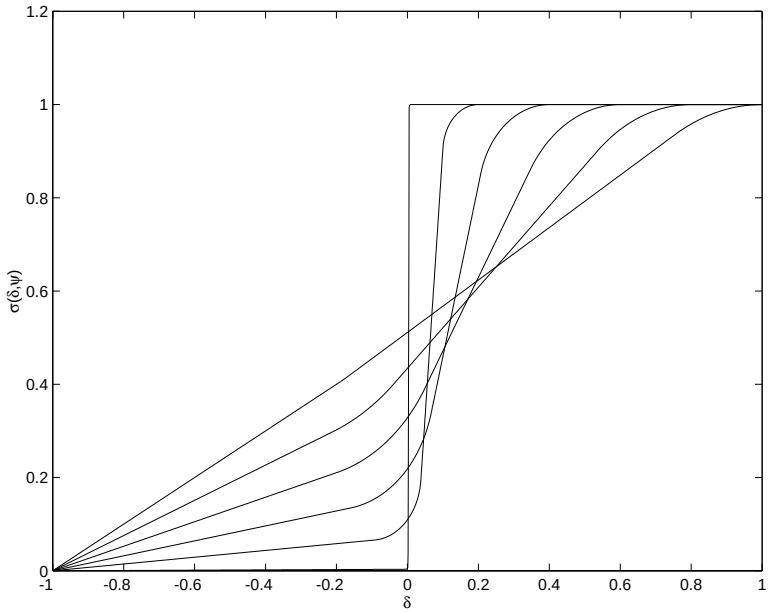


Figure B.1: The synthetic CFM objective function

Assuming once again that the class priors are unknown, the strategy we wish to investigate involves maximizing

$$\min_c \mathrm{E}_{\mathbf{X}|\Omega}[\sigma(\delta(\mathrm{X}|\theta),\psi)|\omega_c] = \min_c \int \sigma(\delta_c(\mathrm{X}|\theta),\psi)\rho(\mathrm{X}|\omega_c)\mathrm{dX}. \tag{B.19}$$

As in the previous section, we expect the maximization process to produce a partition which satisfies

$$\begin{aligned}\int \sigma(\delta_1(\mathrm{X}|\theta),\psi)\rho(\mathrm{X}|\omega_1)\mathrm{dX} &= \int \sigma(\delta_2(\mathrm{X}|\theta),\psi)\rho(\mathrm{X}|\omega_2)\mathrm{dX} = \ldots \\ &= \int \sigma(\delta_{\mathcal{C}}(\mathrm{X}|\theta),\psi)\rho(\mathrm{X}|\omega_{\mathcal{C}})\mathrm{dX}. \end{aligned}\tag{B.20}$$

Our objective now is to determine under what conditions is the minimax condition

$$\int_{\overline{R}_1} \rho(\mathrm{X}|\omega_1)\mathrm{dX} = \int_{\overline{R}_2} \rho(\mathrm{X}|\omega_2)\mathrm{dX} = \ldots = \int_{\overline{R}_{\mathcal{C}}} \rho(\mathrm{X}|\omega_{\mathcal{C}})\mathrm{dX} \tag{B.21}$$

satisfied.

In order to address this question, we need to reexamine properties of the synthetic CFM objective function $\sigma(\delta,\psi)$. $\sigma(\delta,\psi)$ is a monotonic function of the discriminant differential δ. The shape of the objective function varies between a step function and a line over the domain [-1,1] as the confidence parameter ψ increases from 0 to 1. When $\delta > \psi$, $\sigma(\delta,\psi) = 1$. Figure B.1 illustrates the various forms of CFM for a range of confidence parameters.

First let us consider the trivial case when $\psi = 0$. $\sigma(\delta,0) = u(\delta)$ where $u(x)$ is the Heaviside step function. Substituting into equation B.20, we find

$$\begin{aligned}\int \sigma(\delta_i(\mathrm{X}|\theta),\psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX} &= \int u(\delta_i(\mathrm{X}|\theta))\rho(\mathrm{X}|\omega_i)\mathrm{dX} \\ &= \int_{R_i} \rho(\mathrm{X}|\omega_i)\mathrm{dX}. \end{aligned}\tag{B.22}$$

Since

$$\int_{\overline{R}_i} \rho(\mathrm{X}|\omega_i)\mathrm{dX} = 1 - \int_{R_i} \rho(\mathrm{X}|\omega_i)\mathrm{dX}, \tag{B.23}$$

we see that the minimax condition is indeed satisfied. This is no surprise since when $\psi = 0$, $\sigma(\delta_i, 0)$ is simply an indicator function for the decision region R_i associated with the given class ω_i.

Unfortunately, when the objective function is a step function, we cannot apply gradient descent approaches to learn a minimax classifier. Therefore we move now to the more interesting scenario where $\psi > 0$. Consider partitioning the integral

$$\int \sigma(\delta_i(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX} \tag{B.24}$$

into two integrals over disjoint regions of the feature space $\mathbf{X}$. The first integral will be over the region $R_{\delta_i > \psi}$ in $\mathbf{X}$ where $\sigma(\delta, \psi) = 1$. The second integral will be over the complement of this region. This yields

$$\begin{aligned}\int \sigma(\delta_i(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX} = \\ \int_{R_{\delta_i > \psi}} \rho(\mathrm{X}|\omega_i)\mathrm{dX} + \int_{\overline{R}_{\delta_i > \psi}} \sigma(\delta_i(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX}.\end{aligned} \tag{B.25}$$

Examining this decomposition, we notice first of all that the minimax condition is trivially satisfied if the class-conditional densities are separable. In other words, if every example in the feature space $\mathbf{X}$ maps to only one class label, there exists a classifier such that $\delta_i(\mathrm{X}|\theta) > \psi$ for all examples associated with ω_i. Therefore

$$\int_{\overline{R}_{\delta_i > \psi}} \sigma(\delta_i(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX} = 0 \tag{B.26}$$

and the minimax condition is satisfied.

In the majority of classification problems we encounter, the class-conditional densities overlap. So our primary interest is in the general case where no assumptions are placed upon the distributions. Judicious exploitation of the confidence parameter ψ during the maximization of

$$\min_c \int \sigma(\delta_c(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_c)\mathrm{dX} \tag{B.27}$$

will be critical to approach the minimax solution admitted by the hypothesis class which has the minimum worst case error rate.

The learning procedure we will investigate in the future involves annealing the confidence parameter ψ during the optimization process. Beginning with $\psi = 1$, we will maximize the minimum class-conditional CFM until the value converges. Then we will reduce confidence and repeat the optimization process. Our objective is to maximize the number of examples with differentials $\delta > \psi$. In this way we are minimizing the contribution of the integral

$$\int_{\overline{R}_{\delta_i > \psi}} \sigma(\delta_i(\mathrm{X}|\theta), \psi)\rho(\mathrm{X}|\omega_i)\mathrm{dX} \tag{B.28}$$

to equation B.25 and converging on a minimax solution. In the scenario where an infinite amount of training data is available, ψ can be reduced to zero so that we are guaranteed to

obtain a minimax solution. Yet with a finite amount of data, confidence reduction beyond a certain level is unwarranted since a small fraction of the training data will determine the placement of the decision boundaries. Cross-validation will be important for avoiding overfitting of the training data.

Appendix C

Logistic Linear Surfaces of Constant Discriminant Differential

Given the largest discriminant function $g_{(1)}(X|\theta_{(1)}, \theta_{b_{(1)}})$ and the next largest discriminant function $g_{(2)}(X|\theta_{(2)}, \theta_{b_{(2)}})$ corresponding to the vectorized image X, the discriminant differential $\delta(X|\theta)$ is defined as

$$\begin{aligned} \delta(X|\theta) &= g_{(1)}(X|\theta_{(1)}, \theta_{b_{(1)}}) - g_{(2)}(X|\theta_{(2)}, \theta_{b_{(2)}}) && \text{(C.1)} \\ &= \frac{1}{1 + e^{-\theta_{(1)}^T X - \theta_{b_{(1)}}}} - \frac{1}{1 + e^{-\theta_{(2)}^T X - \theta_{b_{(2)}}}} && \text{(C.2)} \end{aligned}$$

where

$$\theta = \left[\theta_{(1)}\ \theta_{b_{(1)}}\ \theta_{(2)}\ \theta_{b_{(2)}}\right]. \tag{C.3}$$

The *principal hyperplanes* associated with the discriminant functions are defined as

$$\theta_{(1)}^T X + \theta_{b_{(1)}} = 0 \tag{C.4}$$

and

$$\theta_{(2)}^T X + \theta_{b_{(2)}} = 0. \tag{C.5}$$

Our objective is to show that the surfaces of constant discriminant differential become parallel to the nearest principal hyperplane as one moves along a surface of constant discriminant differential away from the principal hyperplane intersection. We will demonstrate this by showing that as one moves along a path parallel to a principal hyperplane away from the principal hyperplane intersection, the gradient of the discriminant differential becomes normal to the hyperplane.

We begin by computing the gradient of the discriminant differential $\nabla_X \delta(X|\theta)$ with respect to the vectorized image X. We find

$$\begin{aligned} \nabla_X \delta(X|\theta) &= \frac{\partial}{\partial X}\left[\frac{1}{1 + e^{-\theta_{(1)}^T X - \theta_{b_{(1)}}}} - \frac{1}{1 + e^{-\theta_{(2)}^T X - \theta_{b_{(2)}}}}\right] && \text{(C.6)} \\ &= \frac{e^{-\theta_{(1)}^T X - \theta_{b_{(1)}}}}{\left(1 + e^{-\theta_{(1)}^T X - \theta_{b_{(1)}}}\right)^2}\theta_{(1)} - \frac{e^{-\theta_{(2)}^T X - \theta_{b_{(2)}}}}{\left(1 + e^{-\theta_{(2)}^T X - \theta_{b_{(2)}}}\right)^2}\theta_{(2)}. && \text{(C.7)} \end{aligned}$$

Simplifying the exponential terms, we obtain

$$\begin{aligned}\nabla_X \delta(X|\theta) &= \frac{1}{2 + e^{\theta_{(1)}^T X + \theta_{b_{(1)}}} + e^{-\theta_{(1)}^T X - \theta_{b_{(1)}}}} \theta_{(1)} - \\ &\quad \frac{1}{2 + e^{\theta_{(2)}^T X + \theta_{b_{(2)}}} + e^{-\theta_{(2)}^T X - \theta_{b_{(2)}}}} \theta_{(2)} \qquad \text{(C.8)} \\ &= \frac{1}{2\left(1 + \cosh\left(\theta_{(1)}^T X + \theta_{b_{(1)}}\right)\right)} \theta_{(1)} - \\ &\quad \frac{1}{2\left(1 + \cosh\left(\theta_{(2)}^T X + \theta_{b_{(2)}}\right)\right)} \theta_{(2)}. \qquad \text{(C.9)}\end{aligned}$$

Since $\cosh(X)$ is an even function,

$$\nabla_X \delta(X|\theta) = \frac{1}{2\left(1 + \cosh\left(|\theta_{(1)}^T X + \theta_{b_{(1)}}|\right)\right)} \theta_{(1)} - \frac{1}{2\left(1 + \cosh\left(|\theta_{(2)}^T X + \theta_{b_{(2)}}|\right)\right)} \theta_{(2)}. \qquad \text{(C.10)}$$

In order to fully understand this equation, let us consider the geometric interpretation of the function

$$h_i(X) = |\theta_{(i)}^T X + \theta_{b_{(i)}}|. \qquad \text{(C.11)}$$

Given the vector X_0 that is parallel or antiparallel to $\theta_{(i)}$ and satisfies

$$\theta_{(i)}^T X_0 + \theta_{b_{(i)}} = 0, \qquad \text{(C.12)}$$

$h_i(X)$ can be expressed as

$$\begin{aligned} h_i(X) &= |\theta_{(i)}^T X - \theta_{(i)}^T X_0| && \text{(C.13)} \\ &= \|\theta_{(i)}\| \, \|X - X_0\| \, |\cos(\gamma)| && \text{(C.14)} \\ &= \|\theta_{(i)}\| \, d_{(i)}(X) && \text{(C.15)}\end{aligned}$$

where γ is the angle between $\theta_{(i)}$ and $X - X_0$. This form indicates $h_i(X)$ is the distance $d_{(i)}(X)$ from X to the principal hyperplane weighted by the magnitude of $\theta_{(i)}$. Therefore equation C.10 can be written as

$$\nabla_X \delta(X|\theta) = \frac{1}{2\left(1 + \cosh(\|\theta_{(1)}\| \, d_{(1)}(X))\right)} \theta_{(1)} - \frac{1}{2\left(1 + \cosh(\|\theta_{(2)}\| \, d_{(2)}(X))\right)} \theta_{(2)} \qquad \text{(C.16)}$$

which provides an intuitively appealing result. Now consider moving along a path parallel to the principal hyperplane defined by $[\theta_{(1)} \ \theta_{b_{(1)}}]$ away from the principal hyperplane intersection. This implies $d_{(1)}(X)$ remains constant while $d_{(2)}(X)$ tends toward infinity which drives the corresponding scale factor for $\theta_{(2)}$ toward zero. So in the limit,

$$\nabla_X \delta(X|\theta) = \frac{1}{2\left(1 + \cosh(\|\theta_{(1)}\| \, d_{(1)}(X))\right)} \theta_{(1)} \qquad \text{(C.17)}$$

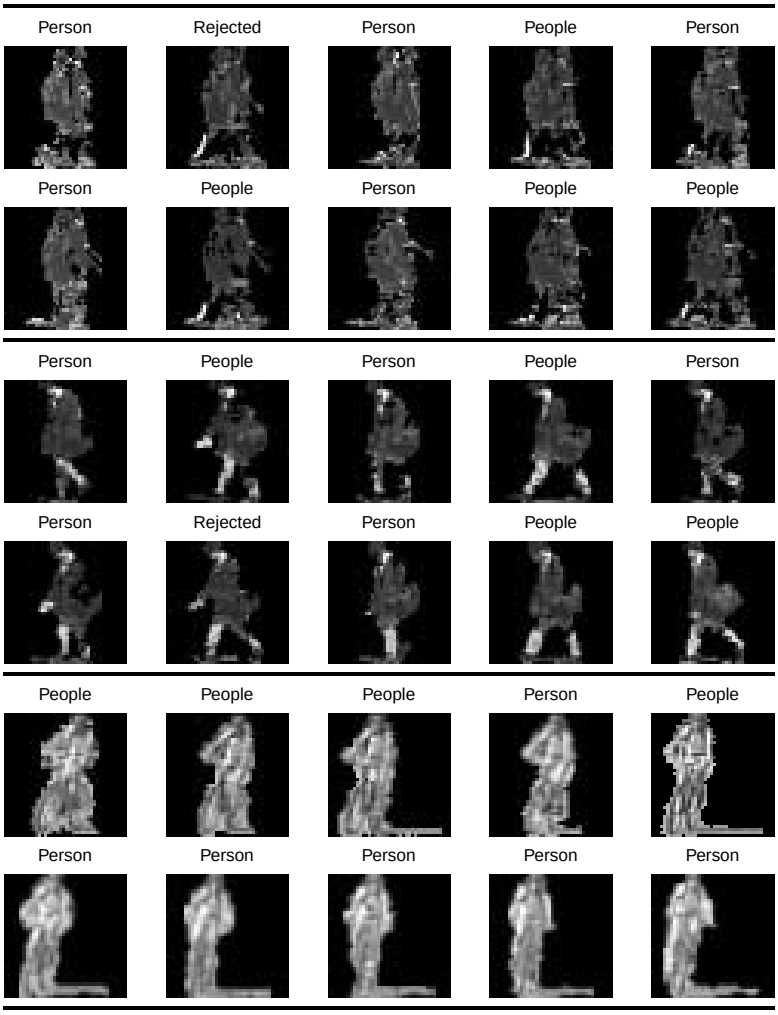
which indicates the surface of constant discriminant differential becomes parallel to the principal hyperplane as the distance from the principal hyperplane intersection increases. The same argument can be made for the case where one moves along a path parallel to the principal hyperplane defined by $[\theta_{(2)} \ \theta_{b_{(2)}}]$.

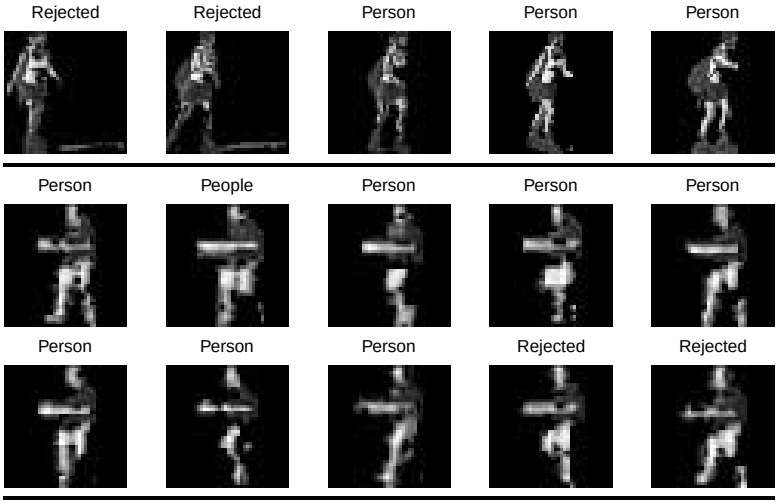
Appendix D

Sorted Image Sequences

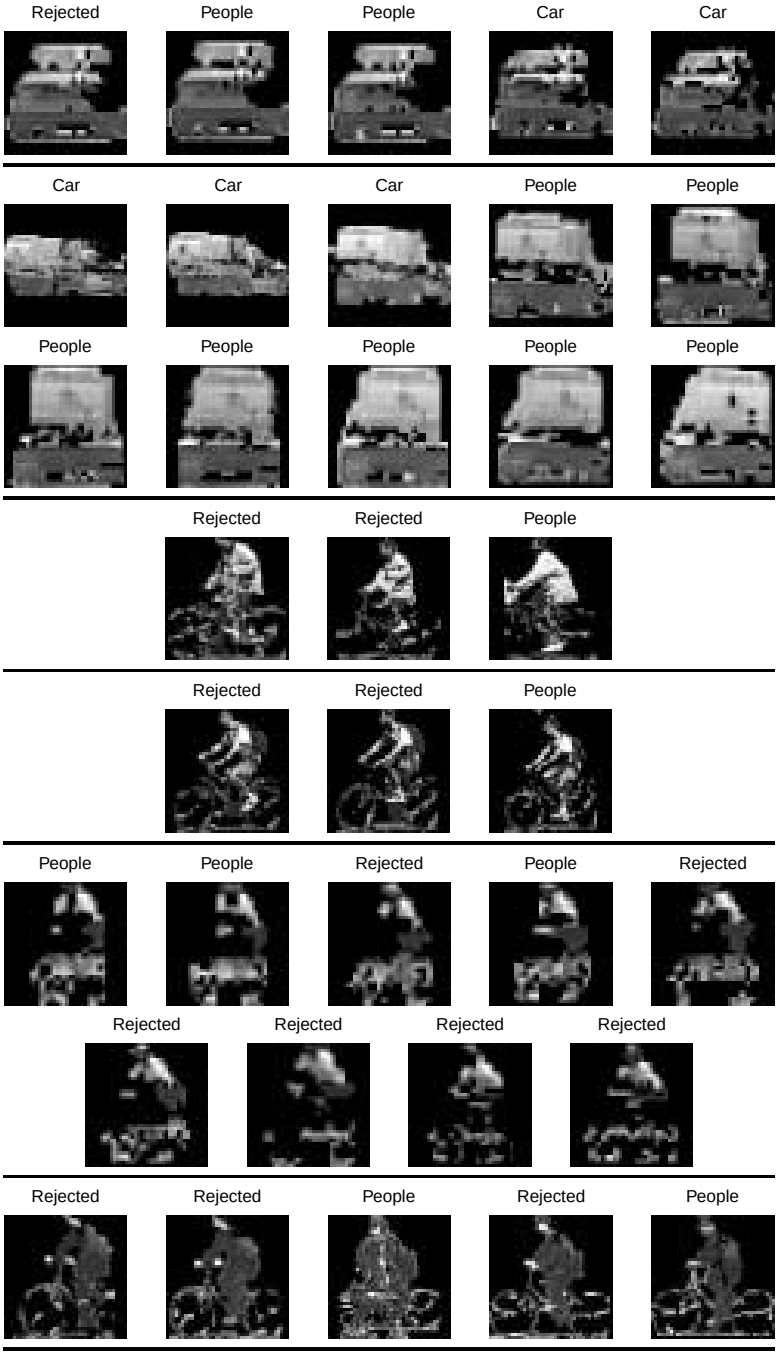
Below are a series of image sequences from the dataset used to evaluate the utility of the classification process for novel image sequence detection. Each set of image sequences is presented in the order of increasing discriminant differential.

Example Image Sequences Labeled as Person

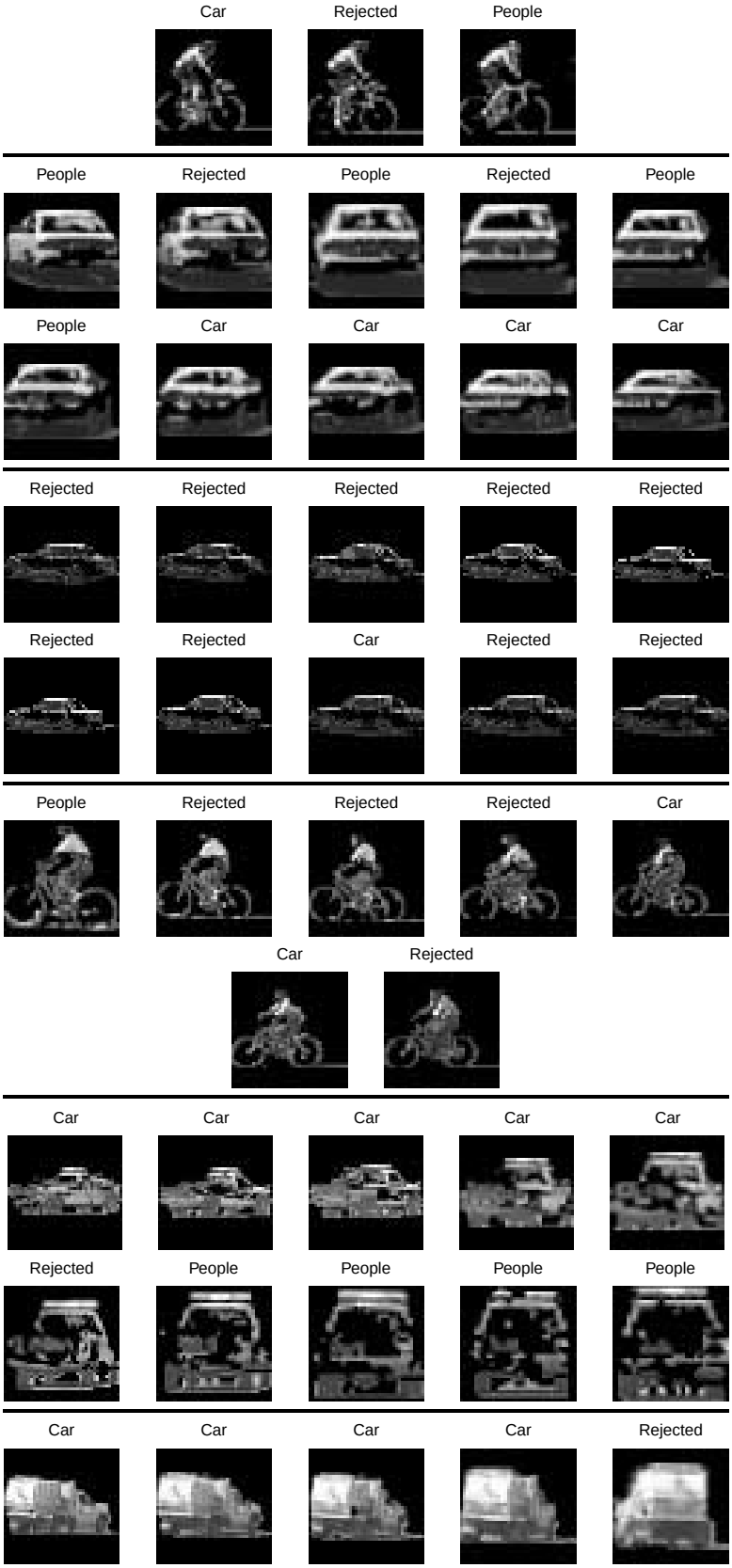


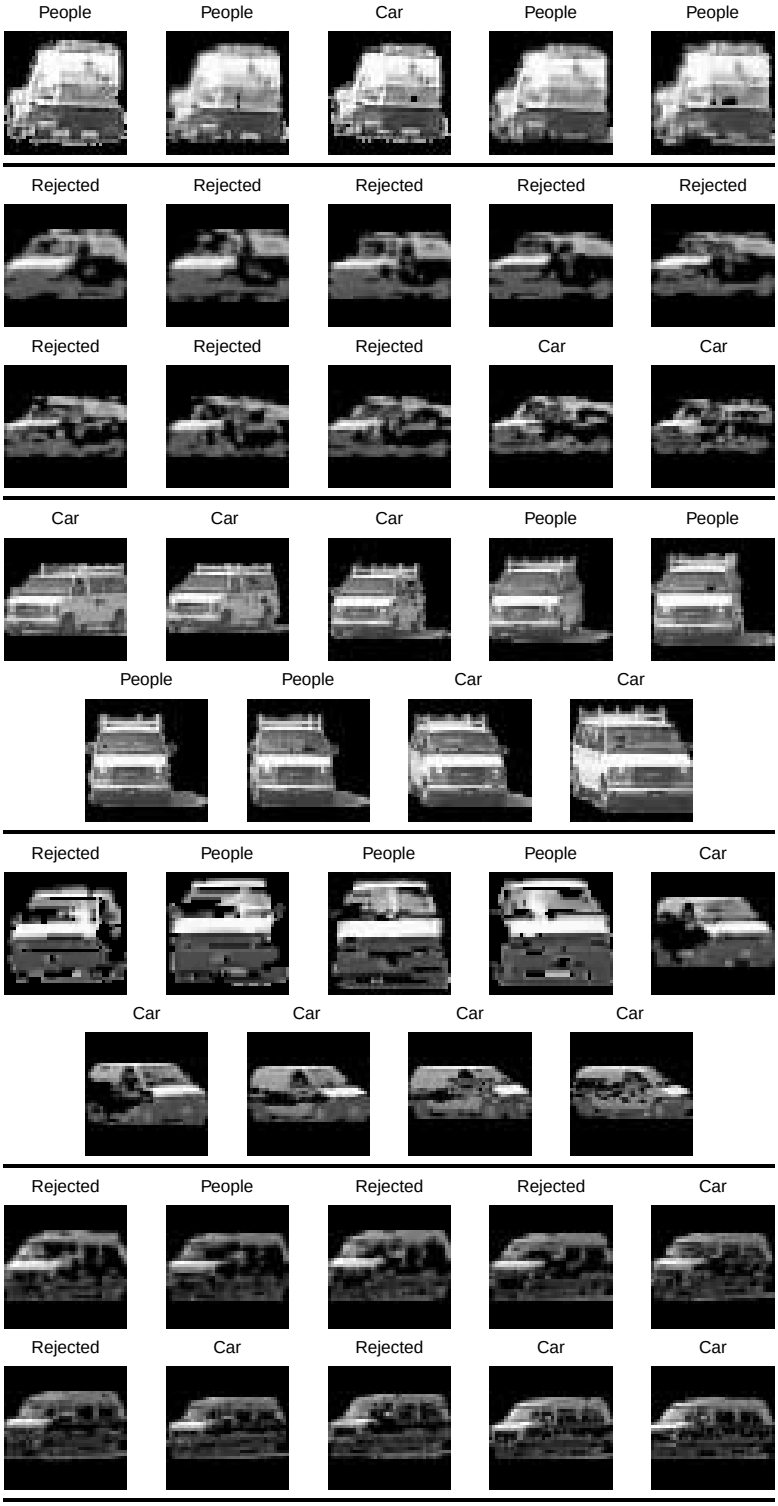

Rejected
Rejected
Person
Person
Person
Person
People
Person
Person
Person
Person
Person
Person
Rejected
Rejected

Example Image Sequences Labeled as People


Rejected
People
People
Car
Car
Car
Car
Car
People
People
People
People
People
People
People
People
Rejected
Rejected
People
Rejected
Rejected
People
People
People
Rejected
People
Rejected
Rejected
Rejected
Rejected
Rejected
Rejected
Rejected
People
Rejected
People

Example Image Sequences Labeled as Car




People
People
Car
People
People
Rejected
Rejected
Rejected
Rejected
Rejected
Rejected
Rejected
Rejected
Car
Car
Car
Car
Car
Car
People
People
People
People
Car
Car
Rejected
People
People
People
Car
Car
Car
Car
Car
Rejected
People
Rejected
Rejected
Car
Rejected
Car
Rejected
Car
Car

Appendix E

Thesis Defense Discussion

This section addresses two discussion points from the thesis defense.

Image Normalization and Differential Maximization

A question was raised about the validity of the approach of selecting the image translation that maximizes the differential. This concern stemmed from the belief that the maximum differential provides a potentially misleading measure of the margin. It is true that a large differential does not necessarily imply that the example is classified with confidence when normalizing based on the differential. Several translations may produce differentials of nearly equal magnitude that correspond to different class labels. Therefore a minor change in the image may produce a change in the class label in such situations, regardless of the magnitude of the maximum differential.

In order to properly measure separability, we should redefine the measure of the margin. One suggested definition was the following. Instead of normalizing the image such that the differential is maximized, consider maximizing the output of each discriminant function over the range of possible translations prior to evaluating the differential. Mathematically this amounts to computing the differential

$$\delta(X|\theta) = \tilde{g}_{(1)}(X|\theta) - \tilde{g}_{(2)}(X|\theta) \tag{E.1}$$

based upon the output from the discriminant functions

$$\tilde{g}_k(X|\theta) = \max_n g_k(\mathcal{T}(X,n)|\theta). \tag{E.2}$$

Preliminary experimental results indicate that this measure does not offer improvements in either classification or rejection performance. Further investigation is needed to fully address this issue.

Bounding the Worst Case Performance

One committee member expressed concern about the effect of minimizing an upper bound on the worst case error rate. Recall that the worst case error rate is defined as the maximum class-conditional error rate. Given the class-conditional error rate estimates were based on relatively small samples, we chose to compute an upper bound on each class-conditional error rate estimate and select the maximum upper bound as our worst case bound. If minimizing the Hoeffding bound leads to poor performance, one may need to employ a more sophisticated measure of uncertainty. A margin-based bound should provide a better characterization of classifier performance than the Hoeffding bound which is only a function of the sample size.

Bibliography

[1] Shlomo Argamon-Engelson and Ido Dagan. Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360, 1999.

[2] K. Bennett and O. L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3:27–39, 1993.

[3] K. P. Bennett and J. A. Blue. A support vector machine approach to decision trees. In *Proceedings, IEEE International Joint Conference on Neural Networks*, pages 2396–2401, 1998.

[4] Kristin P. Bennett, Nello Cristianini, John Shawe-Taylor, and Donghui Wu. Enlarging the margins in perceptron decision trees. Submitted to *Machine Learning*, 1999.

[5] Chris M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] Aaron F. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society, Series B, (Biological Sciences)*, 352(1358):1257–1265, 1997.

[7] Marc Bogaert, Nicolas Chleq, Philippe Cornez, Carlo Regazzoni, Andrea Teschioni, and Monique Thonnat. The PASSWORDS project. In *Proceedings, International Conference on Image Processing*, pages 675–678, 1996.

[8] Chris J. C. Burges. Simplified support vector decision rules. In *Proceedings, Thirteenth International Conference on Machine Learning*, pages 71–77, 1995.

[9] Chris J. C. Burges and David J. Crisp. Uniqueness of the SVM solution. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*. Morgan Kaufmann, 2000.

[10] Chris J. C. Burges and Bernhard Schölkopf. Improving the accuracy and speed of support vector machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 375–381. MIT Press, 1997.

[11] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[12] Hilary Buxton and Shaogang Gong. Advanced visual surveillance using Bayesian networks. In *Proceedings, IEEE Workshop on Context-Based Vision*, 1995.

[13] Colin Campbell. An introduction to kernel methods. To appear in Radial Basis Function Networks: Design and Applications, R.J. Howlett and L.C. Jain, editors, Springer Verlag, Berlin, 2000.

[14] Colin Campbell, Nello Cristianini, and Alex Smola. Query learning with large margin classifiers. To appear in Proceedings of ICML 2000, Stanford, CA.

[15] H. Chernoff. A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[16] Nicolas Chleq and Monique Thonnat. Realtime image sequence interpretation for video surveillance applications. In *Proceedings, International Conference on Image Processing*, pages 801–804, 1996.

[17] D. Corrall. VIEWS: Computer vision for surveillance applications. In *Proceedings, IEE Colloquium on Active and Passive Techniques for 3-D Vision*, pages 8/1–3, 1991.

[18] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[19] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[20] Yann Le Cun, John S. Denker, and Sara A. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. Morgan Kaufmann, 1990.

[21] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[22] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag New York, Inc., 1996.

[23] Christopher P. Diehl, Mahesh Saptharishi, John B. Hampshire II, and Pradeep K. Khosla. Collaborative surveillance using both fixed and mobile unattended ground sensor platforms. *Proceedings of the SPIE*, 3713:178–185, July 1999.

[24] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc., 1973.

[25] Bruce E. Flinchbaugh and Thomas J. Olson. Autonomous video surveillance. *Proceedings of the SPIE*, 2962:144–151, 1997.

[26] G. L. Foresti. A real-time system for video surveillance of unattended outdoor environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(6):697–704, October 1998.

[27] G. L. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1045–1062, October 1999.

[28] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[29] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2–3):133–168, August-September 1997.

[30] Marco Gori and Franco Scarselli. Are multilayer perceptrons adequate for pattern recognition and verification? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1121–1132, November 1998.

[31] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[32] Isabelle Guyon, Nada Matić, and Vladimir Vapnik. Discovering informative patterns and data cleaning. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 181–203. AAAI Press/MIT Press, 1996.

[33] Ismail Haritaoglu, Ross Cutler, David Harwood, and Larry S. Davis. Backpack: Detection of people carrying objects using silhouettes. In *Proceedings, Seventh IEEE International Conference on Computer Vision*, pages 102–107, 1999.

[34] Ismail Haritaoglu, David Harwood, and Larry S. Davis. Ghost: A human body part labeling system using silhouettes. In *Proceedings, 14th International Conference on Pattern Recognition*, pages 77–82, 1998.

[35] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W^4: Who? When? Where? What? A real-time system for detecting and tracking people. In *Proceedings, Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 222–227, 1998.

[36] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W^4S: A real-time system for detecting and tracking people in $2\frac{1}{2}D$. In *Proceedings, Fifth European Conference on Computer Vision*, pages 877–892, 1998.

[37] Ismail Haritaoglu, David Harwood, and Larry S. Davis. Active outdoor surveillance. In *Proceedings, Tenth International Conference on Image Analysis and Processing*, pages 1096–1099, 1999.

[38] Ismail Haritaoglu, David Harwood, and Larry S. Davis. Hydra: Multiple people detection and tracking using silhouettes. In *Proceedings, Tenth International Conference on Image Analysis and Processing*, pages 280–285, 1999.

[39] Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In T. J. Sejnowski, G. E. Hinton, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 5*, pages 164–171. Morgan Kaufmann, 1993.

[40] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[41] Berthold Klaus Paul Horn. *Robot Vision*. MIT Press, 1986.

[42] J. B. Hampshire II and Barak A. Pearlmutter. Equivalence proofs for multi-layer perceptron classifiers and the Bayesian discriminant function. In Touretzky, Elman, Sejnowski, and Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, pages 159–172. Morgan Kaufmann, 1991.

[43] J. B. Hampshire II and A. H. Waibel. A novel objective function for improved phoneme recognition using time delay neural networks. In *Proceedings, International Joint Conference on Neural Networks*, volume 1, pages 235–241, June 1989.

[44] John B. Hampshire II. *A Differential Theory of Learning for Efficient Statistical Pattern Recognition*. PhD thesis, Carnegie Mellon University, September 1993.

[45] John B. Hampshire II and Alex Waibel. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Transactions on Neural Networks*, 1(2):216–228, June 1990.

[46] Y. A. Ivanov and A. F. Bobick. Recognition of multi-agent interaction in video surveillance. In *Proceedings, Seventh International Conference on Computer Vision*, pages 169–176, 1999.

[47] Takeo Kanade, Robert T. Collins, Alan J. Lipton, Peter Burt, and Lambert Wixson. Advances in cooperative multi-sensor video surveillance. In *Proceedings, DARPA Image Understanding Workshop*, 1998.

[48] Michael Kearns and Leslie G. Valiant. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aiken Computation Laboratory, 1988.

[49] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[50] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *Proceedings, 11th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

[51] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of 17th International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 3–12. Springer-Verlag, 1994.

[52] Alan J. Lipton, Hironobu Fujiyoshi, and Raju S. Patil. Moving target classification and tracking from real-time video. In *Proceedings, DARPA Image Understanding Workshop*, 1998.

[53] David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1992.

[54] Llew Mason, Peter Bartlett, and Jonathan Baxter. Direct optimization of margins improves generalization in combined classifiers. In *Advances in Neural Information Processing Systems 11*, pages 288–294. MIT Press, 1999.

[55] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Functional gradient techniques for combining hypotheses. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[56] A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume XII, pages 615–622, Polytechnic Institute of Brooklyn, 1962.

[57] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones. A multi-agent framework for visual surveillance. In *Proceedings, Tenth International Conference on Image Analysis and Processing*, pages 1104–1107, 1999.

[58] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings, International Conference on Computer Vision*, pages 555–562, 1998.

[59] Constantine P. Papageorgiou and Tomaso Poggio. A pattern classification approach to dynamical object detection. In *Proceedings, International Conference on Computer Vision*, volume 2, pages 1223–1228, 1999.

[60] Constantine P. Papageorgiou and Tomaso Poggio. Trainable pedestrian detection. In *Proceedings, International Conference on Image Processing*, volume 4, pages 35–39, 1999.

[61] John C. Platt, Nello Cristianini, and John Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*. Morgan Kaufmann, 2000.

[62] Gunnar Rätsch, Bernhard Schölkopf, Alexander J. Smola, Sebastian Mika, Takashi Onoda, and Klaus-Robert Müller. Robust ensemble learning. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 207–219. MIT Press, 1999.

[63] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *Proceedings, Sixth International Conference on Computer Vision*, pages 857–862, January 1998.

[64] Stephen J. Roberts. Assessing the confidence of classification in artificial neural networks. In *Proceedings, IEE Colloquium on Artificial Intelligence Methods for Biomedical Data Processing*, pages 4/1–4/6, 1996.

[65] Danny Roobaert. Improving the generalization of linear support vector machines: an application to 3D object recognition with cluttered background. In *Proceedings, Support Vector Machine Workshop at the 16th International Joint Conference on Artificial Intelligence*, pages 857–862, August 1999.

[66] Danny Roobaert. View-based 3D object recognition with support vector machines. In *Proceedings, IEEE Workshop on Neural Networks for Signal Processing*, pages 77–84, August 1999.

[67] Frank Rosenblatt. *Principles of Neurodynamics: Perceptron and Theory of Brain Mechanisms*. Spartan Books, Washington D.C., 1962.

[68] Mahesh Saptharishi. Assessing feature relevance online using differential discriminative diagnosis. Master's thesis, Carnegie Mellon University, 1999.

[69] Mahesh Saptharishi, John B. Hampshire II, and Pradeep K. Khosla. Agent-based moving object correspondence using differential discriminative diagnosis. In *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 652–658, June 2000.

[70] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[71] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(3):1651–1686, 1998.

[72] Henry Schneiderman and Takeo Kanade. A statistical model for 3D object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.

[73] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Robert C. Williamson, and Alex J. Smola. Estimating the support of a high-dimensional distribution. Technical Report 99-87, Microsoft Research, 1999.

[74] Bernhard Schölkopf, Alex J. Smola, Robert Williamson, and Peter Bartlett. New support vector algorithms. Technical Report NC2-TR-1998-031, NeuroCOLT2 Technical Report Series, November 1998.

[75] Bernhard Schölkopf, Robert C. Williamson, Alex Smola, and John Shawe-Taylor. SV estimation of a distribution's support. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*. Morgan Kaufmann, 2000.

[76] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Workshop on Computational Learning Theory*, pages 287–294, 1992.

[77] Robert E. Shapire. A brief introduction to boosting. In *Proceedings, Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[78] John Shawe-Taylor. Confidence estimates of classification accuracy on new examples. In *Proceedings, Third European Conference on Computational Learning Theory*, pages 260–271, March 1997.

[79] Michael E. Tipping. The relevance vector machine. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 652–658. Morgan Kaufmann, 2000.

[80] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[81] Vladimir V. Vapnik. *Statistical Learning Theory*. Springer-Verlag New York, 1998.

[82] Vladimir V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 2000.

[83] J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings, Seventh European Symposium on Artificial Neural Networks*, pages 219–224, 1999.