Joint Deconvolution and Classification: Classifiers for Dataset Shift Induced by Linear Systems

Hyrum S. Anderson

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2010

Program Authorized to Offer Degree: Department of Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Hyrum S. Anderson

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

_____

Maya R. Gupta

Reading Committee:

_____

Les E. Atlas

_____

Maya R. Gupta

_____

John D. Sahr

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Joint Deconvolution and Classification:
Classifiers for Dataset Shift Induced by Linear Systems

Hyrum S. Anderson

Chair of the Supervisory Committee:
Professor Maya R. Gupta
Department of Electrical Engineering

A basic assumption underlying traditional supervised learning algorithms is that labeled examples used to train a classifier are indicative of (drawn i.i.d. from the same distribution as) the test sample. However, a common problem in signal processing violates this assumption: given clean training examples, classify a signal that has propagated through a noisy linear time-invariant system. This traditional signal processing problem is recast as a dataset shift problem for machine learning, in which training and test distributions differ. Joint deconvolution and classification is proposed as a system-optimized framework for classifying a channel-corrupted signal from clean training features. In particular, classifiers are designed to account for the convolution relationship between test and training distributions. The joint MAP classifier jointly estimates a clean signal and a class label from a multipath-corrupted signal. The joint QDA classifier probabilistically accounts for the convolution relationship, and is extended for use with subband energy features. A set of kernels are proposed that measure similarity between a clean training signal and a corrupted test signal, and their use for channel-robust SVMs is proposed. With a focus on passive acoustic classification for multipath-corrupted signals, classifiers are tested in experiments to classify simulated narrowband acoustic signals, to identify Bowhead whales from their vocalizations in shallow water, and to acoustically identify trumpeters in a reverberant environment.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

> Perhaps the greatest gift a teacher can give to a student is the sense of complete enjoyment that comes from immersion in the subject being taught.
>
> Dr. Henry Eyring, chemist

The start of my studies at UW coincided with other significant changes. Nicole and I left our careers in Boston to move to Seattle. Our first son was born during midterms of my first quarter, and our second followed two-and-a-half years later. First and foremost I thank Nicole and our two sons, Scott and James, for their support in this life-changing adventure. Having my family to come home to during graduate school brought purpose to my studies and richness to my life.

Additionally, I cannot discount the influence of my father and mother who have offered a lifetime of encouragement and determination to overcome challenges. Likewise, my wife's parents and family have shown unfeigned interest and support in my studies; it's been a great comfort to live in close proximity to them.

I am thankful for the mentorship provided by my adviser, Maya Gupta. She introduced me to key concepts in this thesis, partnered in the development of the many ideas, and encouraged my exploration in others. I admire the investment she places in her students, providing us opportunities for ownership, leadership, scholarship, and friendship. Because of this, her research lab has attracted the best colleagues that a student can hope for.

Additional thanks to my committee, to friends in the Information Design Lab, and to Dr. Henry Eyring—whose quotable wisdom begins each chapter—for shaping this thesis.

Chapter 1

DATASET SHIFT FROM LINEAR TIME-INVARIANT SYSTEMS

> Like any other honest man, I am
> obliged to accept only the truth.
>
> Dr. Henry Eyring, chemist

Classification methods predict the class membership of a test feature vector $\mathbf{x}$ given a set of labeled training feature vectors $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ (see Figure 1.1). For instance, in handwritten digit classification the vector $\mathbf{x}_i$ may be a vector of pixel intensities of the ith example image, the label y_i signifies the digit represented by the image, and a classifier must determine which digit is represented by the pixel intensities of $\mathbf{x}$. For most supervised learning algorithms, the underlying assumption is that the training samples $\{\mathbf{x}_i\}_{i=1}^N$ are similar to any new test sample $\mathbf{x}$ that might be observed, so that an algorithm trained on $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ can be used to make a reasonable prediction about the label of $\mathbf{x}$.

However, there are many applications in which the training samples differ significantly from the test sample. For example, automatic speech recognition systems may be trained in a certain acoustic environment, but the environment at test time is not guaranteed to match the training conditions [8, 54]. Or, consider an underwater acoustics scenario, in which training samples of target are acquired in the deep ocean water, but the classifier is deployed in a shallow-water environment, where the test samples are corrupted by multipath and interference [2, 44]. As another example, face recognition methods must sometimes account for the fact that a test image was acquired from a low-quality camera, but the database of training images is high-resolution [26]. Each of these examples violates the default assumption in supervised learning, that a test sample $\mathbf{x}$ and its true label y^* are drawn i.i.d. from the same underlying joint distribution p_{XY} as the training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. A mismatch between the test and training distributions is known as *dataset shift* [53].

If the training distribution p_{XY} differs from the test distribution arbitrarily, then one

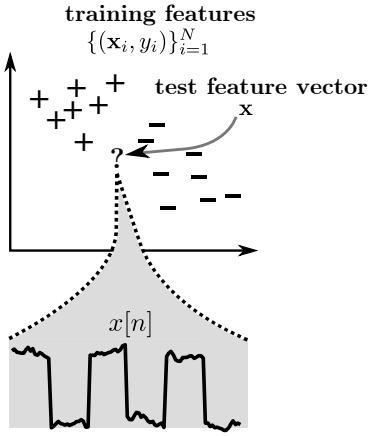


Figure 1.1: Typical supervised learning classification setup. The task is to classify a feature vector $\mathbf{x}$ extracted from the signal $x[n]$ given labeled ($+$ or $-$) training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The test feature $\mathbf{x}$ and its true label are assumed to be independent and identically distributed with $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$.

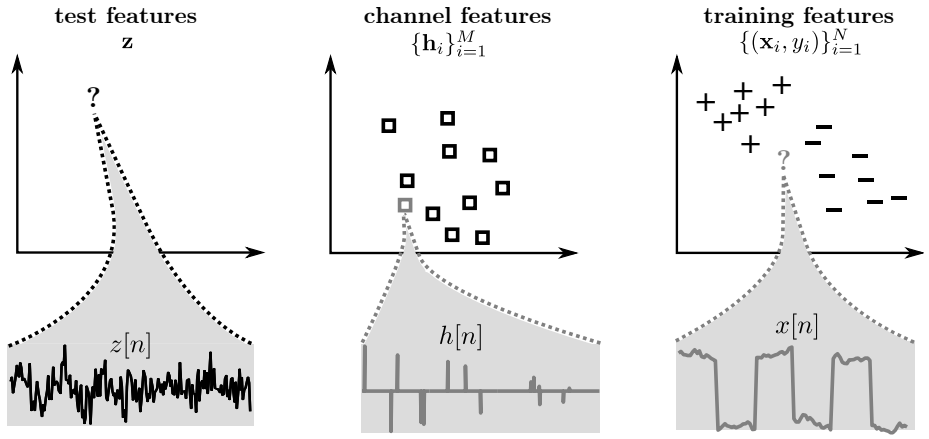


Figure 1.2: Depiction of problem setup for dataset shift induced by a linear time-invariant system. The task is to classify feature vector $\mathbf{z}$ extracted from the signal $z[n]$. Features $\mathbf{x}$ of the unknown test signal $x[n]$ are not given, however $\mathbf{x}$ and its true label y^* are drawn i.i.d. from same distribution p_{XY} as training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Likewise, the unknown feature vector $\mathbf{h}$ of the true impulse response $h[n]$ is independent and identically distributed with auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$.

cannot hope to learn a classifier from training data that generalizes well to the test data. However, each of the examples above arises from a case where the training samples and test samples are related by an unknown linear time-invariant system and additive noise.

This thesis addresses the dataset shift problem induced by a noisy time-invariant channel. Formally, training samples $\{\mathbf{x}_i\}_{i=1}^N$ are extracted from critically sampled time signals $\{x_i[n]\}_{i=1}^N$, and a test feature vector $\mathbf{z}$ is extracted from the signal

$$z[n] = h[n] * x[n] + w[n], \tag{1.1}$$

where $*$ denotes convolution, $h[n]$ is the unknown impulse response of the unknown system, $x[n]$ is the unknown signal of interest, and $w[n]$ is a noise realization. Features $\mathbf{x}$ of the unknown test signal $x[n]$ are not known, however, it is assumed that $\mathbf{x}$ and its true label y^* are drawn i.i.d. from the same distribution as the training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. It is assumed that the labels $\{y_i\}_{i=1}^N$ represent one of two classes, $y_i = -1$ or $y_i = 1$. In addition, it is assumed that a finite set of unlabeled auxiliary samples $\{\mathbf{h}_i\}_{i=1}^M$ are available, and that the unknown features $\mathbf{h}$ of the $h[n]$ in (1.1) are drawn i.i.d. from the same joint distribution as $\{\mathbf{h}_i\}_{i=1}^M$. (Throughout this thesis, bold-face $\mathbf{x}$ denotes a vector, regular-face x denotes a scalar; random vectors and scalars are uppercase and written as $\mathbf{X}$ and X, respectively; refer to Table 1.1 for a summary of the key notation used in this thesis.)

The feature vectors $\mathbf{z}$, $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\mathbf{h}_i\}_{i=1}^M$ may, for example, consist of samples of a signal, pixel intensities, wavelet coefficients, cepstral coefficients, or the energy in selected frequency bins. No matter the choice of features, it is clear that in general $p_{ZY} \neq p_{XY}$: the test distribution has shifted from the training distribution because of the noisy linear time-invariant channel.

1.1 Scope of Research

The novel aspects of this thesis lie at the fruitful intersection of signal processing and machine learning. This thesis presents a machine learning setup (see Figure 1.2) to a well-known problem in signal processing, and therein formally defines a previously unpublished flavor of dataset shift. Then, several classifiers are presented to address the dataset shift induced by a noisy time-invariant system.

Table 1.1: Notation used in this thesis.

$x[n]$	discrete-time signal
$\mathbf{x}$	generic feature vector (discrete-time signal vector where noted)
$x^f[k]$	DFT of $x[n]$
$\mathbf{x}^f$	DFT signal vector
$u_x[k]$	subband energy $\lvert x^f[k]\rvert^2$
$\mathbf{u}_x$	subband energy vector
$\mathbf{X}$	random vector
$\bar{\mathbf{X}}$	mean of $\mathbf{X}$
N	# of training samples
M	# of auxiliary channel examples
d	# of feature dimensions
$\mathcal{N}(\mathbf{x};\mathbf{m},A)$	Gaussian in $\mathbf{x}$ with mean $\mathbf{m}$ and covariance A
$\mathbf{a}*\mathbf{b}$	discrete convolution sum of the vectors $\mathbf{a}$ and $\mathbf{b}$
$A**B$	two-dimensional convolution of matrices A and B
$A\cdot B$	Hadamard (elementwise) multiplication: $C=A\cdot B \leftrightarrow C_{ij}=A_{ij}B_{ij}$
$\frac{[A]}{[B]}$	Hadamard division: $C=\frac{[A]}{[B]} \leftrightarrow C_{ij}=\frac{A_{ij}}{B_{ij}}$

There are four general strategies for dealing with the problem outlined in Figure 1.2. First, one may select features $\mathbf{z}$ and $\{\mathbf{x}_i\}_{i=1}^N$ that are invariant to the LTI channel, so that $p_{ZY} = p_{XY}$. Features invariant to channel dispersion were considered for classification in [50]. A second strategy is to estimate $\hat{\mathbf{x}}$ from $\mathbf{z}$ via feature or signal deconvolution, so $\hat{\mathbf{x}}$ and its label are jointly distributed as the training distribution p_{XY}. Since classifiers are trained on the original dataset, distributed as p_{XY}, these methods are informally referred to as *x-space classifiers*. A third approach is to transform the training data (explicitly or implicitly) using $\{\mathbf{h}_i\}_{i=1}^M$ to yield artificial training pairs $\{(\mathbf{z}_i, y_i)\}_{i=1}^P$ that are distributed i.i.d. with the test distribution p_{ZY}. Methods that utilize p_{ZY} as the underlying distribution are informally referred to as *z-space classifiers*. Yet a fourth method is to train a classifier using p_H as the underlying distribution: for each test point $\mathbf{z}$, labeled channel estimates $\{(\hat{\mathbf{h}}_i, y_i)\}_{i=1}^N$ are estimated from $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the label y^* is chosen by comparing test points $\{\hat{\mathbf{h}}_i\}_{i=1}^N$ to $\{\mathbf{h}_i\}_{i=1}^M$ using a one-class classifier. Gupta et al. presented a 1-NN classifier for multipath impulse responses that chose $y^* = y_\ell$ for $\hat{\mathbf{h}}_\ell$ that maximized impulsiveness [28].

In contrast to classical approaches in signal processing that seek to deconvolve then classify in distinct operations, the methods presented in this thesis represent a system-optimized approach to the problem, and may be categorized broadly under *joint deconvolution and classification*. It should come as no surprise that if we are given training samples $\{\mathbf{x}_i\}_{i=1}^N$ and auxiliary samples $\{\mathbf{h}_i\}_{i=1}^M$ as shown in Fig. 1.2, one might better estimate $\hat{x}[n]$ from $z[n]$ than by employing a blind deconvolution method that ignores the given training and auxiliary data. From the perspective of building a classifier, one may expect that better performance can be achieved if the convolution relationship between $x[n]$ and $z[n]$ is built into a classifier. A central conclusion of this thesis confirms this intuition: in order to estimate the class for $z[n]$, one need not perform deconvolution explicitly; instead, better results can be achieved when a classifier is designed to account for the linear time-invariant channel.

The linear-system dataset shift problem depicted in Figure 1.2 describes many applications, but the experiments in this thesis will principally address passive acoustic classification, particularly when $h[n]$ in equation (1.1) is the impulse response of a multipath

channel, and $w[n]$ is such that the SNR of the received signal is very low—between -10 and 10 dB SNR. Multipath channels exist whenever there is more than a single propagation path between a transmitter and receiver. Since the transmitted signal travels multiple paths of different propagation lengths, the resulting phase shifting and attenuation of individual paths combines constructively or destructively at the receiver. The multipath impulse response depends on the source location, receiver location and the location and acoustic impedance of all scatterers in the propagation channel, so $h[n]$ is not a smooth function of position [39].

Since a dominant effect in multipath channels are "echoes" of various delays, $h[n]$ is often modeled as being k-sparse, that is, $h[n]$ contains only k nonzero elements whose location represents the phase delay and whose amplitude represents the attenuation of a given echo path. However, the location and amplitude of nonzero elements cannot be specified without accurate knowledge of the propagation environment. The difficulty of modeling multipath effects is exacerbated for ocean acoustic channels [64], depicted in Figure 1.3. First, the speed of sound is a function of temperature, salinity and pressure, so that it varies non-monotonically with depth, but as a rule of thumb is about 1560 m/s—more than four times faster than in dry air. In addition, the multipath impulse response depends on surface interactions that vary with wind speed (surface) and sediment composition (bottom). Research related to classifying acoustic signals in shallow water is cited in Chapter 2.

1.2 Background

This section provides the reader with background related to joint deconvolution and classification. It is not intended to be exhaustive, rather, it places the research presented in this thesis in proper context, and establishes basic tools and notation that subsequent chapters will draw upon.

1.2.1 Blind Deconvolution

Much research in signal and image processing, applied math and seismology has been applied to the problem of deconvolution, both blind and non-blind. Non-blind deconvolution is in general an ill-posed problem, since the convolution with impulse response $h[n]$ may have a

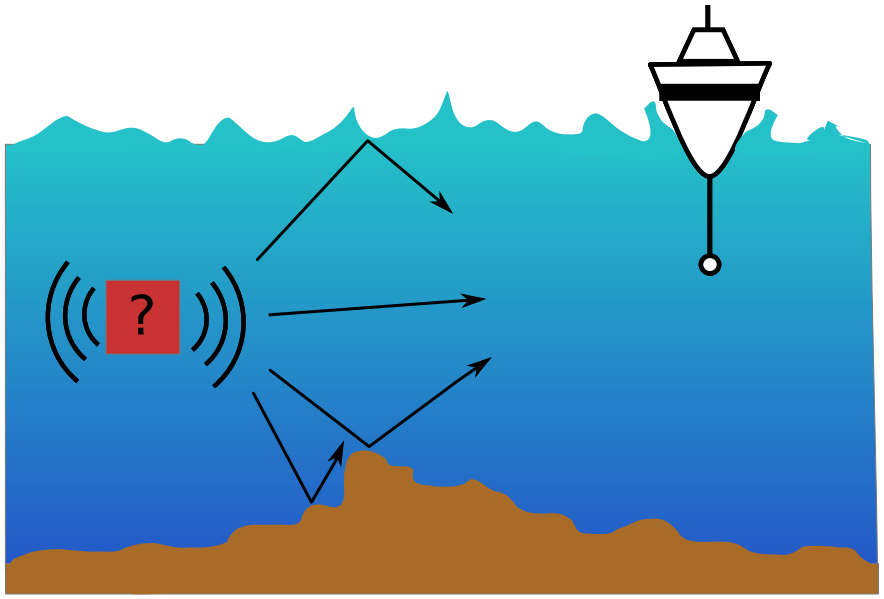


Figure 1.3: Underwater passive acoustic classification is complicated in shallow-water environments by multipath channels. Water may be considered shallow when the ocean depth is shallowed compared to the observation distance. The multipath impulse response depends on source and receiver location, surface interactions that vary with wind speed (surface) and sediment composition (bottom), and the sound speed profile of the water column.

non-trivial null-space. The additive noise $w[n]$ exacerbates the problem. Thus, even when $h[n]$ is specified exactly, a signal $\hat{x}[n]$ estimated from $z[n]$ and $h[n]$ is not a unique solution to $z[n] = h[n] * x[n] + w[n]$.

Blind deconvolution methods attempt to recover $\hat{x}[n]$ (or $\hat{h}[n]$) from $z[n]$ without specifying $h[n]$ (respectively, $x[n]$). Instead, they rely on prior information about the structure of the signal that is being recovered. There are a host of blind deconvolution algorithms for applications ranging from communications to geophysics. For example, Petropulu and Nikias presented a cepstral blind deconvolution technique that requires that there are no pole-zero cancelations between the signal $x[n]$ and the channel impulse responses from two measurements [52]; however, the method requires multiple observations and relies on cepstral methods that are too sensitive for the low SNR regimes that we are interested in. Signal-based deconvolution methods to remove multipath have been considered in [9, 7, 67, 65]. These methods rely on the fact that $h[n]$ may be modeled using only a few nonzero coefficients.

As an example of an approach applied to deconvolving multipath-corrupted signals, consider *minimum entropy blind deconvolution* introduced by Cabrelli in the geophysics community [9]. Cabrelli introduces the so-called D-norm that operates on a discrete impulse response (vector $\mathbf{h}$):

$$D(\mathbf{h}) \triangleq \frac{\|\mathbf{h}\|_\infty}{\|\mathbf{h}\|_2}, \tag{1.2}$$

where $\|\mathbf{h}\|_\infty = \max_n |h[n]|$ and $\|\mathbf{h}\|_2$ is the ℓ_2 norm. Since $\|\cdot\|_\infty \leq \|\cdot\|_2$, $D(\mathbf{h})$ achieves a maximum of 1, which can be shown to occur when $\mathbf{h}$ has exactly one nonzero element. Thus, $D(\mathbf{h})$ measures the "impulsiveness" of $\mathbf{h}$. Let $\mathcal{F}_\ell$ be the set of all nonzero vectors of fixed length ℓ—the set $\mathcal{F}_\ell$ is the set of all ℓ-length FIR filters. Let $\mathbf{z} \in \mathbb{R}^{n+\ell-1}$ be the (noiseless) discrete convolution $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{h} \in \mathcal{F}_\ell$. Cabrelli's method solves

$$\mathbf{g} = \arg\max_{\mathbf{f} \in \mathcal{F}_\ell} D(\mathbf{f} * \mathbf{z}). \tag{1.3}$$

If the unknown $\mathbf{h}$ uniquely satisfies the *minimum entropy* property that $D(\mathbf{h}) > D(\mathbf{f}), \forall \mathbf{f} \in \{\mathcal{F}_\ell \setminus \mathbf{h}\}$, then $\mathbf{g} = \mathbf{x}^{-1}$ is the Fourier inverse of $\mathbf{x}$. This can be verified by noting that when $D(\mathbf{h})$ is the unique maximum of $D(\cdot)$ and since $\mathbf{x}^{-1} * \mathbf{z} = \mathbf{x}^{-1} * \mathbf{x} * \mathbf{h} = \mathbf{h}$, then $D(\mathbf{g} * \mathbf{z}) = D(\mathbf{h})$ implies that $\mathbf{g} = \mathbf{x}^{-1}$.

1.2.2 Classifiers and Kernels

Supervised learning classifiers infer the class label, assumed to be either $+1$ or -1, of a test feature vector $\mathbf{x}$ from labeled training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, as depicted in Figure 1.1. Classifiers in this thesis are of two general varieties: generative and discriminative. For a full review of classifiers, the reader is referred to [30].

Generative Classifiers and QDA

Generative classifiers estimate from training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ the joint distribution $p(\mathbf{x}, y)$, factorized as the class generating distribution and prior $p(\mathbf{x}|y)p(y)$. Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are two common generative classifiers that classify a test sample $\mathbf{x}$ with each class-conditional distribution $p(\mathbf{x}|y)$ assumed to be Gaussian $\mathcal{N}(\mathbf{x}; \mathbf{m}_y \Sigma_y)$ with the same covariance matrix $\Sigma_{y=-1} = \Sigma_{y=1}$ (LDA)

for each class y is, or allowing covariance matrices to differ (QDA). The learning task, then, is to estimate $\mathbf{m}_y$ and Σ_y from $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, which may be ill-posed for $d > N$. The decision boundaries resulting from LDA and QDA are, respectively, linear and quadratic (or any conic section), hence their names. Generative classifiers' probabilistic nature provides straightforward extensions for handling priors and missing data, and can readily diagnose if the test sample $\mathbf{x}$ is ill-fitted to either class. However, a criticism of generative classifiers is that since one does not know the true distribution p_{XY}, choosing a simple model for $p(\mathbf{x}|y)$ introduces model bias, while choosing a too-flexible model for $p(\mathbf{x}|y)$ leads to overfitting (exaggerating the importance of some data that may be noisy or irrelevant).

Discriminative Classifiers, the SVM, and Kernels

Discriminative approaches eschew modeling of p_{XY}, instead directly optimizing the decision boundary by minimizing misclassifications (empirical risk) over the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Hence, they have the perceived advantage of directly optimizing the quantity of interest (classification error via empirical risk), instead of the description of the classes through $p(\mathbf{x}|y)$. In practice, they have been shown to be robust, even when few training samples are available [35]. However, confidence measures are difficult to determine from discriminative classifiers, and prior knowledge is difficult to incorporate—both of which come naturally to generative classifiers.

The support vector machine (SVM) is currently the most popular member of the family of discriminative classifiers. The SVM classifies a feature vector $\mathbf{x}$ based on the sign of a discriminant function $f(\mathbf{x})$ as $y^* = \text{sgn}(f(\mathbf{x}))$. The SVM can be motivated from the viewpoint of a maximal margin linear classifier that solves for weights $\mathbf{v}$ and bias b of a linear function $f(\mathbf{x}) = \mathbf{v}^T\mathbf{x} + b$ as

$$\min_{\mathbf{v},\xi_i,b} \frac{1}{2}\mathbf{v}^T\mathbf{v} + C\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad y_i\left(\mathbf{v}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N,$$

where C is a regularization parameter often expressed as $C = \frac{1}{2\lambda N}$. When the classes are linearly separable (all slack variables $\xi_i = 0$), the SVM finds the function $f(\mathbf{x})$ so that among all linear classifiers that bisect the data via $\text{sgn}(f(\mathbf{x}))$, the margin $\frac{2}{\|\mathbf{v}\|}$ is maximized.

Often, a linear decision boundary is too inflexible. The SVM is easily extended to nonlinear decision boundaries by introducing a nonlinear mapping $\phi : \mathbb{R}^d \mapsto \mathbb{R}^p$ and solving the SVM on the mapped data $\{(\phi(\mathbf{x}_i), y_i)\}_{i=1}^N$. This can be accomplished implicitly by embedding $\phi(\cdot)$ into the optimization and reformulating as

$$\min_{\mathbf{c},\xi_i,b} \frac{1}{2}\mathbf{c}^T K \mathbf{c} + C\sum_{i=1}^N \xi_i \quad \text{s.t.} \quad y_i \left(\sum_{j=1}^N c_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N,$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is the kernel function, K is the kernel matrix whose i, jth element is $K(\mathbf{x}_i, \mathbf{x}_j)$, and each weight $c_i = \alpha_i y_i$ is a multiple of the class label $y_i \in \{-1, +1\}$, where $0 \leq \alpha_i \leq C$. The fact that $\phi(\cdot)$ appears only as an inner product $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ is called the *kernel trick*: rather than specifying $\phi(\cdot)$, one may instead specify a kernel function $K(\cdot, \cdot)$ which implies an underlying $\phi(\cdot)$.

The kernel-based discriminant function can then be expressed as

$$f(\mathbf{x}) = b + \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i). \tag{1.4}$$

Notably, the SVM often chooses many $\alpha_i = 0$, so that (1.4) depends only on a sparse subset of the training data. Hence the name of the classifier—those $\mathbf{x}_i$'s for which $0 < \alpha_i < C$ are the *support vectors* of the SVM. For any support vector[1] $\mathbf{x}_j$, the function $f(\mathbf{x}_j)$ exactly predicts its label y_j, so that the bias $b = y_j - \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j)$.

The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ measures the similarity of its two arguments. Since it implicitly represents the inner product $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, it must be a symmetric positive definite function. The family of kernels is a convex cone, so that one may construct new kernels from existing kernels using a pleasant algebra [24]. We exploit this fact when deriving channel-robust kernels in Chapter 5.

Radial basis function kernels can be written in terms of some distance between training points $K(\mathbf{x}_i, \mathbf{x}_j) = K(\|\mathbf{x}_i - \mathbf{x}_j\|)$, hence they are invariant to a global translation of the dataset. In Chapter 5, we focus on the popular Gaussian radial basis function (RBF):

$$K_{\text{rbf}}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}\left(\mathbf{x}_i; \mathbf{x}_j, \gamma^{-1} I\right), \tag{1.5}$$

[1] Those $\mathbf{x}_j$'s for which $\alpha_j = C$ are *bounded support vectors*, and do *not* satisfy $b = y_j - \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j)$.

where $\mathcal{N}(\cdot)$ denotes the Gaussian function, and γ is the bandwidth parameter. Typically, the RBF is implemented without the Gaussian scaling factor so that $K_{\text{rbf}}(\mathbf{x}, \mathbf{x}) = 1$.

Every kernel is associated with a reproducing kernel Hilbert space (RKHS). This fact allows us to express the SVM objective more elegantly as minimizing regularized loss:

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(f(\mathbf{x}_i), y_i) + \lambda \|f\|_K^2, \tag{1.6}$$

where the hinge loss $L(f(\mathbf{x}), y) = (1 - yf(\mathbf{x}))_+$ is a convex relaxation of 0/1 loss, and $\mathcal{H}$ is the RKHS associated with kernel K.

1.2.3 Features

The classifiers presented in this thesis may be generalized for many choices of features, but the development is restricted to two important types of features in signal processing. In some applications (e.g., image classification), it is convenient to train a classifier using the sampled signal (e.g., pixels) as features. Let $\mathbf{x}$, $\mathbf{h}$, $\mathbf{w}$ and $\mathbf{z}$ be vectors whose elements contain the samples of the discrete-time signals $x[n]$, $h[n]$, $w[n]$ and $z[n]$, respectively. Then, the convolution relationship in (1.1) can be expressed using the notation

$$\mathbf{z} = \mathbf{h} * \mathbf{x} + \mathbf{w},$$

where $\mathbf{a} * \mathbf{b}$ is the vector of values that results from discrete convolution of the entries in $\mathbf{a}$ with the entries in $\mathbf{b}$.

In other applications, features extracted from the discrete-time signals better discriminate the different classes. Subband energy features are a useful and frequently utilized feature choice in many signal processing applications. Let $x^f[k]$ denote the kth bin of the discrete Fourier transform of $x[n]$, and let $w^f[k]$ be a realization of a zero-mean proper complex Gaussian white noise process with known variance. The subband energy of $u_z[k] = \left|z^f[k]\right|^2$ is given by

$$u_z[k] = u_h[k]u_x[k] + u_w[k] + 2\,\text{Re}\left\{x^f[k]h^f[k]w^{f*}[k]\right\},$$

where $w^{f*}[k]$ is the complex conjugate of $w^f[k]$. Consider a feature vector of subband energies at d frequency bins $k = k_1, k_2, \ldots, k_d$. The relationship of the observed vector

$\mathbf{u}_z \in \mathbb{R}^d$ and the (unknown) vector $\mathbf{u}_x \in \mathbb{R}^d$ can be written concisely as

$$\mathbf{u}_z = \mathbf{u}_h \cdot \mathbf{u}_x + \mathbf{u}_w + 2\,\mathrm{Re}\left\{\mathbf{x}^f \cdot \mathbf{h}^f \cdot \mathbf{w}^{f*}\right\}, \tag{1.7}$$

where $\cdot$ denotes the Hadamard (element-wise) product.

1.2.4 Noisy Features

If instead of a convolution relationship, we have only an additive noise model $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{w}$, then a test feature vector $\tilde{\mathbf{x}}$ differs from the training samples $\{\mathbf{x}_i\}_{i=1}^N$ only by additive noise. This simple noise model can be handled by the *noisy features* rule, which accounts for some or all of the elements of $\tilde{\mathbf{x}}$ to be noisy or missing [17, p. 55]. When all of the elements are noisy, the noisy features *maximum a posteriori* (noisy features MAP) rule is

$$p(y|\tilde{\mathbf{x}}) \propto \int p(\mathbf{x}|y)p(y)p(\tilde{\mathbf{x}}|\mathbf{x})\, d\mathbf{x},$$

where the pdfs in the integrand are assumed to be known.

1.2.5 Dataset Shift

Dataset shift refers to the general problem in predictive modeling in which the joint distribution of inputs and outputs differs between training and test stages. As mentioned previously, if training and test distributions differ arbitrarily, there is little hope for learning a generalizable classifier.

This thesis defines one particular type of dataset shift that is induced by an unknown linear time-invariant channel. However, dataset shift of other varieties are present in many other applications. For example, since spam filtering algorithms are trained under one set of criteria, successful spammers often build spamming mechanisms that change behavior over time to exploit this fact [53]. Or, consider the sample selection bias problem of training a speech recognition system using predominantly native English speakers, but in real-world test conditions, accents and drawls confuse the classifier. Despite the fact that dataset shift is present to some degree in all practical applications, it has until recently received little direct attention in machine learning.

Recently, algorithms have been developed to deal with a particular type of dataset shift called covariate shift. In covariate shift, only the distribution of the inputs (covariates) $\mathbf{x}$ changes between training and test time, but the input-output relationship remains unchanged [5]: for training, $p_{XY} = p_{Y|X}\, p_X$, but at test time, $\tilde{p}_{XY} = p_{Y|X}\, \tilde{p}_X$.

Prior probability shift is another common type of dataset shift occurs when only the distributions of the outputs y changes between training and test time: for training, $p_{XY} = p_{X|Y}\, p_Y$, but at test time, $\tilde{p}_{XY} = p_{X|Y}\, \tilde{p}_Y$.

In the parlance of dataset shift, the problem addressed in this thesis occurs because of a probabilistic domain shift. Specifically, the test distribution p_{ZY} is related to the training distribution p_{XY} via the conditional distribution $p_{Z|X}$ as $p_{ZY} = \int_X p_{Z|X} p_{XY}$. We model $p_{Z|X}$ using prior knowledge about the problem structure—that the underlying relationship between training and test samples is rooted in the convolution relationship in (1.1)—and by using auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$.

For an overview of other forms of dataset shift and algorithms to cope with them, the reader is referred to [53].

1.3 Outline of Thesis

The thesis proceeds as follows. Chapter 2 presents related research from signal processing and machine learning that are applicable to classifying a signal corrupted by unknown multipath.

Chapters 3 through 5 cover the theoretical development of classifiers. In Chapter 3, two signal-based classifiers are presented. The joint MAP algorithm jointly estimates a class label y^* and deconvolution estimate $\hat{\mathbf{x}}$. The joint QDA algorithm estimates only the class label; the convolution relationship is built into the classifier. In Chapter 4, a feature-based joint QDA algorithm is derived which utilizes subband energy features to discriminate between classes. The local joint QDA algorithm is also introduced as a means to reduce model bias. In Chapter 5, channel-robust kernels are presented which account for the dataset shift induced by linear systems, with closed form derivations given for RBF kernels with Gaussian $p_{Z|X}$ for discrete-time signal, image, and subband energy features. The kernels are used with SVMs in the experiments of Chapter 6.

The experiments in Chapter 6 compare feature-based classifiers on three datasets: the first uses simulated narrowband signals with simulated channel impulse responses, in the second, Bowhead whale vocalizations are used to identify individual whales in a simulated acoustic environment, and lastly, trumpeters are identified from recordings in an acoustically reverberant chamber.

Some insights into related work in x-space classifiers that were developed in parallel with the classifiers in this thesis are presented as part of the conclusions in Chapter 7.

Chapter 2

RELATED WORK

> [Great men are those] who can catch hold of men's minds and feelings and inspire them to do things bigger than themselves....those who stir feelings and imagination and make men struggle toward perfection.
>
> Dr. Henry Eyring, chemist

The work in this thesis builds on previous contributions from two broad fields: signal processing and machine learning. This chapter reviews relevant prior works from both communities. In Section 2.1, research from acoustics and signal processing is cited which relates to classifying signals corrupted by multipath. Then, research from the machine learning community about building invariance into classifiers is presented in Section 2.2.

2.1 Classifying Signals Corrupted by Multipath

Signal processing researchers in underwater passive acoustics have considered the problem of classifying signals corrupted by multipath for over thirty years [58]. Ehrenberg et al. demonstrated in an ocean acoustic propagation experiment that multipath generally cannot be ignored, and that simple time-gating of the received signal can discard too much of the signal information for classification [18, 19]. Multipath induced by a shallow ocean channel presents an additional challenge in that the multipath propagation is generally time-varying so that the structure of $h[n]$ is highly sensitive to spatial location [21, 63].

A review of the literature reveals four general approaches for classifying signals corrupted with multipath. Each approach corresponds roughly to the four general strategies outlined in Section 1.1 for mitigating the dataset shift problem.

The first strategy is to extract features from training signals $\{x_i[n]\}_{i=1}^N$ and the received signal $z[n]$ that are invariant to multipath distortion, then classify based on the multipath-invariant features. Casting this approach into the problem setup defined in Chapter 1, classification methods of this sort seek to find features such that $p_{XY} = p_{ZY}$. Shin et al. consider a number of time-frequency features for clutter rejection [59]. Strausberger et al. compare different distance measures for 1-nearest-neighbor classification of signals passed through Rician channels for over-the-horizon radar [62]. Okopal and Loughlin developed features invariant to channel dispersion and dissipation, and demonstrated superior classification performance compared to temporal and spectral moment features [49]. Other research about features invariant to propagation effects include [50, 51, 45]. In general, classification using channel invariant features can provide good results to the extent that the classes are well-separated in the designated feature space.

Blind deconvolution is the basis for a second commonly-used approach for classifying $z[n]$: a clean signal $\hat{x}[n]$ is estimated from $z[n]$, then a classifier is used on features of $\hat{x}[n]$. By deconvolving, the dataset shift problem is ameliorated since both test and training samples are distributed as p_{XY}. There are many examples of trying to remove multipath by blind deconvolution in order to classify [43, 11, 55, 38, 7, 67, 65]. Some researchers exploit the impulsiveness of the unknown $h[n]$ to estimate $\hat{x}[n]$ via blind deconvolution [7, 67, 65]. Broadhead and Pflug reported [7] excellent correlation between true signals and signals blindly deconvolved by the minimum entropy method using the D-norm [9], but did not consider classification. Gupta et al. have shown that these blind deconvolution estimates can be highly correlated to out-of-class training signals, so that nearest neighbor classification on correlation scores performs poorly, particularly at low signal-to-noise ratios [28, 27].

A third approach is to predict the representation of the training signals $\{x_i[n]\}_{i=1}^N$ using a forward model for the multipath $\hat{h}[n]$. This has the advantage of avoiding deconvolution, and is most related to the joint QDA classifier presented in Chapter 4. A classifier is built using virtual training signals $\{z_i[n] = x_i[n] * \hat{h}[n]\}_{i=1}^N$ to classify $z[n]$, so that both test and training data are distributed as p_{ZY}. Researchers previously have based their forward model $\hat{h}[n]$ on geometry or physical assumptions [43, 15]. Liu et al. first proposed an in-channel classifier based on free-field training data [43]. They built a classifier by assuming

a finite number of multipath reflections for near-bottom target classification. Dasgupta and Carin classify after accounting for multipath via time-reversal imaging, which requires the geometry and sound speed profile of the channel [15].

In a conference paper, we (Gupta et al.) first proposed that to classify a signal corrupted by unknown multipath, jointly considering deconvolution and classification can lead to better performance than traditional approaches that deconvolve then classify in independent steps [28]. Our method leveraged training data to produce a multipath channel candidate $\hat{h}_i[n]$ for each training signal $x_i[n]$ given $z[n]$. Then, a nearest-neighbor classifier chose the class $y^* = y_i$ for which the estimated filter $\hat{h}_i[n]$ was most multipath-like, according to Cabrelli's D-norm in (1.2). The resulting joint deconvolution and classification method yields the best signal estimate $\hat{x}[n] = x_i[n]$ and filter estimate $\hat{h}[n] = \hat{h}_i[n]$ that may have produced $z[n]$, as well as the optimal class label $y^* = y_i$. Classification performance was markedly better than minimum entropy blind deconvolution followed by classification, particularly at low signal-to-noise ratios. However, the performance of that joint deconvolution/classifier relied on several conditions [28]. First, it required a good criterion for evaluating how well a given $\hat{h}[n]$ represented a multipath filter. Although the D-norm criterion is a convenient choice, multipath impulse responses in underwater acoustics violate the minimum entropy property [43]. Secondly, the proposed nearest-neighbor approach required that the training signals $\{x_i[n]\}_{i=1}^N$ be plentiful and that the true $x[n]$ be close to a training sample of the correct class in terms of $\|x[n] - x_i[n]\|$. Thirdly, the deconvolution estimate $\hat{x}[n]$ was always restricted to be a member of the set $\{x_i[n]\}_{i=1}^N$. Lastly, it is not straightforward to incorporate features in classification.

2.2 Invariant Classifiers

In the machine learning community, researchers have addressed the fact that the training samples $\{\mathbf{x}_i\}_{i=1}^N$ may not encapsulate all of the manifestations of a test sample. And, similar to the dataset shift problem outlined in Chapter 1 in which we know *a priori* that test and training data are related by convolution, machine learning researchers have proposed methods to incorporate prior knowledge into classifiers, albeit for different applications. The various methods boil down to two general strategies: creating virtual examples that

model the conditions at test time, and designing classifiers that are invariant to the various manifestations of a test sample.

2.2.1 Virtual Examples

The idea of augmenting a training set with "virtual examples" (VEs) dates back to at least 1990 [1]. For example, to build a handwritten digit classifier that is robust to various rotations, one can augment the original training set with artificial examples of rotated digits [16]. The transformed VEs are included with the original training examples to form an expanded training set. The choice of transformation applied to generate the VEs is based on prior knowledge about the perturbations that may be expected in the test features. Typically, the VEs are generated from a discrete and deterministic set of transformations, for example, single pixel translations in the four principal directions of the image plane.

Lorens et al. have employed virtual examples to train SVMs to classify targets from their acoustic signatures [44]. High quality recorded signatures are artificially corrupted by simulating their propagation through an acoustic channel to produce virtual examples that better represent the test distribution p_{ZY}. Since the original training features are not representative of the test distribution, they are discarded. In Chapter 5, we implement the VE method by propagating each of the N training signals through M example channels, resulting in a training data set size of $M \times N$. This approach has the disadvantage of $O\left(M^3N^3\right)$ complexity in training an SVM.

A variant of VEs is the method of *virtual support vectors*, which trains an SVM on an uncorrupted training set, then generates virtual examples from only the support vectors (and bounded support vectors) [56]. The VSV method has been shown to reduce the overall cost of training an SVM. However, in preliminary results on the datasets used in this thesis, the author found that at least for the RBF kernel—which is known to select many training points as support vectors—the VSV method did not substantially decrease training time, and often exhibited worse performance.

2.2.2 Invariant Classifiers via Kernels

Classifiers can be designed to be invariant to conditions one would expect at test time. Schölkopf et al. showed how to engineer kernel functions that allow SVMs to be invariant under transformations forming a Lie group [57]. They showed that the modified kernel is equivalent to preprocessing the data with a whitening matrix.

An approach that is a hybrid of creating virtual examples and engineering a kernel is *jittering kernels*, in which a set of transformations $\mathcal{T}$ (e.g., single pixel translations of an image an any direction) of a training sample $\mathbf{x}_i$ is considered in the kernel definition [16]. Given a positive definite kernel $K(\cdot, \cdot)$ with $K(\mathbf{x}, \mathbf{x}) = C$ is a constant for any $\mathbf{x}$, the jittering kernel solves

$$K_{\text{jitter}}(\mathbf{x}, \mathbf{x}_i) = \max_{t, \tilde{t} \in \mathcal{T}} K(t(\mathbf{x}), \tilde{t}(\mathbf{x}_i)),$$

so that the jittering kernel measures the maximum similarity over all transformations (jitters) that one might expect at test time. Jittering kernels for SVMs have the advantage over VEs for SVMs in that the jittering kernel SVM scales linearly with the number of transformations $|\mathcal{T}|$, whereas the VE SVM is cubic in the number of transformations. Decoste and Schölkopf employed *jittering kernels* in an SVM to build a classifier robust to slight translations and rotations of handwritten digits and showed previously unmatched error rates on the MNIST benchmark dataset of handwritten digits [16]. Invariant kernels have been further studied by Haasdonk and Burkhardt [29].

Chapter 3

SIGNAL-BASED JOINT DECONVOLUTION AND CLASSIFICATION

> A good model is best, but a bad model is better than nothing.
>
> Dr. Henry Eyring, chemist

This chapter expands on the intuitive idea that by jointly considering the task of deconvolution and classification, better performance may be achieved than if dealing with each task serially. Two classifiers are introduced: the joint MAP classifier, and the joint QDA classifier. Each classifier takes as features the samples of a discrete-time signal features: training samples $\{\mathbf{x}_i\}_{i=1}^N = \{x_i[n]\}_{i=1}^N$, auxiliary channel features $\{\mathbf{h}_i\}_{i=1}^M = \{h_i[n]\}_{i=1}^M$ and test feature vector $\mathbf{z} = z[n]$, where the notation $x[n]$ is overloaded to denote the entire discrete-time signal, not just the signal at the nth location. The contents of this chapter have been published in a journal paper [2] and a conference paper [27].

The framework developed in this chapter will apply to any problem where test and training data are related by the convolution relationship in (1.1), but the applications emphasize the case in which $h[n]$ is the impulse response of an acoustic multipath channel, and where the additive noise $w[n]$ gives rise to low signal-to-noise ratios (SNRs). For passive sonar, $z[n]$ represents the in-channel received signal, $h[n]$ represents the multipath and $x[n]$ is the free-field signal. Underwater multipath channels are generally time-varying and are highly sensitive to spatial location, making them difficult to model effectively [21, 63]. To capture the variability of the channel, the impulse response $h[n]$ will be modeled as a draw of a random process.

In Section 3.1, we unify deconvolution and classification in a joint *maximum a posteriori* (MAP) framework. This method jointly estimates a clean signal $\hat{x}[n]$, a channel estimate $\hat{h}[n]$ and a class label y. In Section 3.2, we argue that if signal estimate $\hat{x}[n]$ is not needed, better classification performance can be achieved by not committing to a particular signal

or channel estimate. We show how a quadratic discriminant analysis (QDA) classifier can be designed to incorporate the effects of uncertain $h[n]$. The algorithms are compared to a deconvolve-then-classify strategy in experiments using simulated multipath channels and signals. A feature-based and a more flexible joint QDA classifier will be presented in Chapter 4. The chapter concludes with a discussion about the methods in Section 3.4.

3.1 Joint MAP Deconvolution and Classification

Let vectors $\mathbf{z}$, $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{w}$ denote the discrete-time test signal, source signal, channel and noise, respectively. In this section, we assume that $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{w}$ are realizations of random vectors $\mathbf{X}$, $\mathbf{H}$ and $\mathbf{W}$, respectively. Let $\mathbf{W} \sim \mathcal{N}(\mathbf{w}; 0, \sigma_w^2 I)$, where I is the identity matrix. It is assumed that $\mathbf{X}$ conditioned on class label y is Gaussian distributed with mean $\mathbf{m}_{x|y}$ and covariance $\Sigma_{x|y}$. Real signals are generally non-Gaussian, but the Gaussian assumption is critical to keeping an otherwise formidable deconvolution problem tractable. The distributions of $\mathbf{X}$, $\mathbf{H}$ and $\mathbf{W}$ are mutually independent. Model $\mathbf{H}$ using a multivariate Laplacian distribution with independent dimensions, so that the ith element of the random multipath has mean $m_h[i]$ and scale parameter $b[i]$. The Laplacian distribution is an appropriate prior model for multipath since it yields sparse realizations. Let $\theta = \{\mathbf{m}_{x|y}, \Sigma_{x|y}, \mathbf{m}_h, \mathbf{b}, \sigma_w^2\}$ be the set of parameters for these three distributions, where θ is assumed to have been estimated *a priori* from the training and auxiliary data.

If the clean signal vector $\mathbf{x}$ were given, the MAP classification rule would select the class label y^* such that

$$y^* = \arg\max_y p(y|\mathbf{x}, \theta). \tag{3.1}$$

However, $\mathbf{x}$ is unknown. One might estimate $\mathbf{x}$ and the channel signal vector $\mathbf{h}$ from $\mathbf{z}$ using the MAP rule as

$$\{\hat{\mathbf{x}}, \hat{\mathbf{h}}\} = \arg\max_{\mathbf{x},\mathbf{h}} p(\mathbf{x}, \mathbf{h}|\mathbf{z}, \theta). \tag{3.2}$$

Instead, we jointly estimate the signal, filter, and class label by combining Equations (3.1) and (3.2) into a single MAP criterion. The proposed joint MAP classifier estimates y^*

as

$$
\begin{aligned}
y^* &\overset{a}{=} \arg\max_y \left(\max_{\mathbf{x},\mathbf{h}} p(\mathbf{x},\mathbf{h},y|\mathbf{z},\theta) \right) && (3.3)\\
&\overset{b}{=} \arg\max_y \left(\max_{\mathbf{x},\mathbf{h}} p(\mathbf{z}|\mathbf{x},\mathbf{h},y,\theta) p(\mathbf{h}|\theta) p(\mathbf{x}|y,\theta) \right) p(y) \\
&\overset{c}{=} \arg\min_y \left[\min_{\mathbf{x},\mathbf{h}} \left(\|\mathbf{z}-\mathbf{h}*\mathbf{x}\|^2 + \sigma_w^2 \|\mathbf{x}-\mathbf{m}_{x|y}\|^2_{\Sigma^{-1}_{x|y}} + 2\sigma_w^2 \sum_i \frac{|h[i]-m_h[i]|}{b[i]} \right) \right] \\
&\qquad + \ \sigma_w^2 \log|\Sigma_{x|y}| - 2\sigma_w^2 \log p(y), && (3.4)
\end{aligned}
$$

where (b) follows from (a) using Bayes' rule, the chain rule and independence assumptions; and (c) follows from (b) by taking the negative logarithm of the pdfs, removing constants that do not depend on $\mathbf{x}$, $\mathbf{h}$ or y from the arg min, and scaling each term by $2\sigma_w^2$. Throughout the thesis, the notation $\|\mathbf{x}\|$ denotes the ℓ_2 norm and $\|\mathbf{x}\|_A^2$ denotes $\mathbf{x}^T A \mathbf{x}$.

The $\|\mathbf{z}-\mathbf{h}*\mathbf{x}\|^2$ term in (3.4) drives the estimated filter $\mathbf{h}$ and test signal $\mathbf{x}$ to be consistent with the received signal $\mathbf{z}$ in terms of squared error. The next two terms in (3.4) drive $\mathbf{x}$ to match the *a priori* expected signal via the ℓ_2 norm, and drive $\mathbf{h}$ to match the *a priori* expected filter via the ℓ_1 norm. Note that these latter two terms are regularized by the noise variance—the greater the noise power, the more the estimate relies on the *a priori* expectations and less on matching the received signal $\mathbf{z}$. The fourth term penalizes classes that exhibit high variance, and the fifth term is the class membership prior. Since the noise determines the degree of regularization, a curious behavior of this approach is that it performs poorly for high SNR: the first term will dominate as σ_w^2 goes to zero, and solutions for $\mathbf{x}$ and $\mathbf{h}$ will no longer depend on $\mathbf{m}_{x|y}$ and $\mathbf{m}_h$, respectively.

The objective function in (3.4) is not convex since it involves a product of variables in the convolution sum $\mathbf{h}*\mathbf{x}$. However, the problem is jointly convex in $\mathbf{x}$ and $\mathbf{h}$ in the limit as $\sigma_w \to \infty$, and is marginally convex in $\mathbf{x}$ or $\mathbf{h}$ for all σ_w. Therefore, we opt to solve the (3.4) using an alternating minimization approach as a heuristic for finding the true solution. Using

$$
H = \text{convmtx}(\mathbf{h}) = \begin{bmatrix} h[1] & \cdots & h[n] & 0 & \cdots & 0 \\ 0 & h[1] & \cdots & h[n] & \cdots & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & h[1] & \cdots & h[n] \end{bmatrix}
$$

for the Toeplitz matrix representation of discrete convolution with fixed $\mathbf{h}$ [33], and for fixed y, the objective as a function of $\mathbf{x}$ can be written in the form of generalized Tikhonov regularization $\|H\mathbf{x} - \mathbf{z}\|^2 + \sigma_w^2 \|\mathbf{h} - \mathbf{m}_h\|^2_{\Sigma_h^{-1}}$. The solution [40] is

$$\hat{\mathbf{x}} = (H^T H + \sigma_w^2 \Sigma_{x|y}^{-1})^{-1} (H^T \mathbf{z} + \sigma_w^2 \Sigma_{x|y}^{-1} \mathbf{m}_{x|y}).$$

Next, solve (3.4) for those terms depending on $\mathbf{h}$ by fixing $\mathbf{x}$ and rewriting as

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \|X\mathbf{h} - \mathbf{z}\|_2^2 + \sigma_w^2 \|D^{-1}(\mathbf{h} - \mathbf{m}_h)\|_1, \tag{3.5}$$

where $X = \text{convmtx}(\mathbf{x})$ is the Toeplitz matrix representation of discrete convolution of $\mathbf{x}$, D is a diagonal matrix with entries $\frac{b[i]}{2}$, and $\|\cdot\|_1$ is the ℓ_1 norm. Equation (3.5) can reformulated as

$$\tilde{\mathbf{h}}^* = \arg\min_{\tilde{\mathbf{h}}} \|\tilde{X}\tilde{\mathbf{h}} - \tilde{\mathbf{z}}\|_2^2 + \sigma_w^2 \|\tilde{\mathbf{h}}\|_1, \tag{3.6}$$

where $\tilde{\mathbf{h}} = D^{-1}(\mathbf{h} - \mathbf{m}_h)$, $\tilde{X} = XD$, and $\tilde{\mathbf{z}} = \mathbf{z} - X\mathbf{m}_h$. Equation (3.6) is solved naturally as a quadratic program with linear constraints, for which efficient algorithms exist [6].

Since the optimization problem in (3.4) is non-convex, the alternating minimizations strategy is not guaranteed to converge to the global minimum [66]. A common approach is to optimize starting from several initial points, then choose the overall minimizer. The initial guesses could be drawn i.i.d. from the class-conditional distribution $\mathcal{N}(\mathbf{m}_{x|y}, \Sigma_{x|y})$. We use a slightly different approach to take advantage of the fact that we have examples from each class: convex combinations of the training signals as initial guesses. A depiction of the alternating minimizations algorithm is shown in Algorithm 1.

Experiments and results for the joint MAP classifier are presented in Section 3.2.

3.1.1 A Related MAP Deconvolution Approach

MAP deconvolution has been explored previously by Lam and Goodman for blind image deblurring (without classification) [42]. In that work, Lam and Goodman estimate the point spread function $\mathbf{h}$ and the image covariance Σ_x by maximizing $p(\mathbf{z}|\mathbf{h}, \Sigma_x)p(\mathbf{h})p(\Sigma_x)$. The prior $p(\Sigma_x)$ is replaced with a heuristic smoothness constraint on the covariance, and the prior $p(\mathbf{h})$ is replaced with the hard constraint $\mathbf{h} \in \mathcal{H}$ for some convex set $\mathcal{G}$. They proposed

Input: $\mathbf{z}$

Output: $\hat{\mathbf{h}}$, $\hat{\mathbf{x}}$, y^*

foreach *class* y **do** initialize($\mathbf{h}_y$);

while *!converged* **do**

  foreach *class* y **do**

    $H = \text{convmtx}(\mathbf{h}_y)$;

    $\mathbf{x}_y = \left(H^T H + \sigma_w^2 \Sigma_{x|y}^{-1}\right)^{-1} \left(H^T z + \sigma_w^2 \Sigma_{x|y}^{-1} \mathbf{m}_{x|y}\right)$;

    $X = \text{convmtx}(\mathbf{x}_y)$;

    $\mathbf{h}_y = \arg\min_h \|X\mathbf{h} - \mathbf{z}\|_2^2 + \sigma_w^2 \left\|D^{-1}\left(\mathbf{h} - \mathbf{m}_h\right)\right\|_1$;

    $\text{score}_y = \|\mathbf{z} - \mathbf{h} * \mathbf{x}\|^2 + \sigma_w^2 \|\mathbf{h} - \mathbf{m}_h\|_{\Sigma_h^{-1}}^2 + \sigma_w^2 \|\mathbf{x} - \mathbf{m}_{x|y}\|_{\Sigma_{x|y}^{-1}}^2 + \sigma_w^2 \log|\Sigma_{x|y}|$;

  end

end

$y^* = \arg\min_y \text{score}_y$;

$\hat{\mathbf{h}} = \mathbf{h}_{y^*}$;

$\hat{\mathbf{x}} = \mathbf{x}_{y^*}$;

Algorithm 1: Joint MAP deconvolution and classification implemented using alternating minimizations. We use $X = \text{convmtx}(\mathbf{x})$ to denote the Toeplitz convolution matrix .

an expectation-maximization (EM) algorithm implementation that alternates between estimating Σ_x (the E-step) and $\mathbf{h}$ (the M-step) in the Fourier domain. The image is finally estimated by Wiener deconvolution using the estimated $\mathbf{h}$ and Σ_x. The algorithm results in high-quality deblurred image estimates [42].

The Lam and Goodman MAP blind deconvolution may be extended to the multipath problem by applying a Laplacian prior for $p(\mathbf{h})$ as we have done in (3.4) instead of their hard constraint $\mathbf{h} \in \mathcal{G}$. However, their approach cannot be extended to fit in the joint deconvolution/classification paradigm by simply conditioning on the class label y and adding the prior $p(y)$ in the optimization. First, Lam and Goodman assume that the image $\mathbf{x}$ (and therefore, $\mathbf{z}$) is a realization of a zero-mean Gaussian distribution. In our framework, the class-conditional mean is an important discriminating feature of the class. Secondly, their

method must estimate Σ_x, but in our framework the class conditional covariance $\Sigma_{x|y}$ is estimated *a priori* from training pairs. Naively replacing Σ_x with $\Sigma_{x|y}$ renders their E-step useless so that iterating does not improve the initial guess. Thus, the training pairs $\{\mathbf{x}_i, y_i\}_{i=1}^N$ offer little advantage to their MAP blind deconvolution technique.

3.2 Probabilistic Deconvolution and Classification Using QDA

Estimating the true signal is difficult and unnecessary if only a class label is required. In this section, we explore classifying signals jointly with *probabilistic* deconvolution, in which a statistical characterization of $x[n]$ and $h[n]$ are used without ever choosing a particular, deterministic signal or channel estimate. Specifically, we consider the maximum likelihood classifier that solves

$$\begin{aligned} y^* &= \arg\max_y p(\mathbf{z}|y) \\ &= \arg\max_y \iint p(\mathbf{z}|\mathbf{x},\mathbf{h},y)p(\mathbf{x}|y)p(\mathbf{h})p(y)\,d\mathbf{x}\,d\mathbf{h}. \end{aligned} \tag{3.7}$$

Assuming a uniform prior, the classifier in (3.7) differs from the joint MAP classifier in (3.3) in that the $\max_{\mathbf{x},\mathbf{h}}$ operator in (3.3) is replaced by expectation over $p(\mathbf{x}|y)$ and $p(\mathbf{h})$ in (3.7). For the remainder of the chapter, we will assume uniform prior $p(y)$ such that $\arg\min_y p(y|\mathbf{z}) = \arg\min_y p(\mathbf{z}|y)$.

Quadratic discriminant analysis (QDA) is a popular classification rule that models each class-conditional distribution in (3.7) as Gaussian [30, 22, 60]. This model can be motivated by the central limit theorem and the fact that the Gaussian is the maximum entropy (least assumptive) distribution given first and second moments. Here, we build a QDA classifier by assuming $p(\mathbf{z}|y)$ in (3.7) is Gaussian, and we show that one can calculate the sufficient statistics $\mathbf{m}_{z|y}$ and $\Sigma_{z|y}$ of $p(\mathbf{z}|y)$ from the estimated mean and covariance of the auxiliary channel features $\{\mathbf{h}_i\}_{i=1}^M$ and the estimated means and covariances of the training signals $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ from each class. Note that we do not make any assumptions on the distributions of $\mathbf{H}$ or $\mathbf{X}$ given y other than that they have finite first and second moments; in fact the result of the convolution $\mathbf{H} * \mathbf{X}$ would not be Gaussian if $\mathbf{H}$ and $\mathbf{X}$ were assumed to be Gaussian random processes.

3.2.1 QDA Classification of Signals Corrupted by LTI Filtering

Let $\mathbf{X}$ be a random vector with finite class-conditional mean $\mathbf{m}_{x|y}$ and finite covariance $\Sigma_{x|y}$; let the noise be a zero-mean random vector $\mathbf{W}$ with covariance $\sigma_w^2 I$; and let $\mathbf{H}$ be a random vector with mean $\mathbf{m}_h$ and covariance Σ_h. Let the observed signal model be given by

$$\mathbf{Z} = \mathbf{X} * \mathbf{H} + \mathbf{W}. \tag{3.8}$$

For Gaussian $p(\mathbf{z}|y)$, compute the class-conditional mean $\mathbf{m}_{z|y}$ and covariance $\Sigma_{z|y}$ of (3.8) as follows:

$$\mathbf{m}_{z|y} = \mathrm{E}\left[\mathbf{X} * \mathbf{H} + \mathbf{W}|y\right] = \mathrm{E}\left[\mathbf{X}|y\right] * \mathrm{E}\left[\mathbf{H}\right] + \mathrm{E}\left[\mathbf{W}\right] = \mathbf{m}_{x|y} * \mathbf{m}_h, \tag{3.9}$$

and

$$\begin{aligned}
\Sigma_{z|y} &\overset{a}{=} \mathrm{E}\left[\left(\mathbf{X} * \mathbf{H} + \mathbf{W}\right)\left(\mathbf{X} * \mathbf{H} + \mathbf{W}\right)^T\right] - \mathbf{m}_{z|y}\mathbf{m}_{z|y}^T \\
&\overset{b}{=} \mathrm{E}\left[\left(\mathbf{X} * \mathbf{H}\right)\left(\mathbf{X} * \mathbf{H}\right)^T\right] + \mathrm{E}\left[\mathbf{W}\mathbf{W}^T\right] - \mathbf{m}_{z|y}\mathbf{m}_{z|y}^T \\
&\overset{c}{=} \mathrm{E}\left[\mathbf{X}\mathbf{X}^T\right] * * \mathrm{E}\left[\mathbf{H}\mathbf{H}^T\right] + \mathrm{E}\left[\mathbf{W}\mathbf{W}^T\right] - \mathbf{m}_{z|y}\mathbf{m}_{z|y}^T \\
&= (\Sigma_{x|y} + \mathbf{m}_{x|y}\mathbf{m}_{x|y}^T) * *(\Sigma_h + \mathbf{m}_h\mathbf{m}_h^T) + \sigma_w^2 I - \mathbf{m}_{z|y}\mathbf{m}_{z|y}^T,
\end{aligned} \tag{3.10}$$

where $**$ denotes two-dimensional discrete convolution. Line (b) follows from (a) since $\mathrm{E}\left[\left(\mathbf{X} * \mathbf{H}\right)\mathbf{W}^T\right] = 0$; line (c) follows from (b) by property (A.1) and independence assumptions; and the expression reduces to (3.10).

The classification rule in (3.7) can be reduced to

$$y^* = \arg\max_y \left(\mathbf{z} - \mathbf{m}_{z|y}\right)^T \Sigma_{z|y}^{-1}\left(\mathbf{z} - \mathbf{m}_{z|y}\right) + \log\left|\Sigma_{z|y}\right|,$$

where the determinant and inverse can both be computed from a single LU factorization of $\Sigma_{z|y}$. This classification approach requires the second-order statistics of $\mathbf{X}$ (conditioned on y) and $\mathbf{H}$ which can be estimated from $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and $\{\mathbf{h}_i\}_{i=1}^M$, respectively. Most blind deconvolution algorithms also require some information or assumptions about either $\mathbf{x}$ or $\mathbf{h}$ or both [42, 7].

3.3 Experiments: Signal-based Joint QDA and Joint MAP Classification

The proposed methods were tested in two experiments that differ in how simulated multipath is generated. In the first, a Laplacian random process was used to generate realizations of multipath channels. In the second, a random k-sparse model was used. In both experiments the clean training signals were drawn from class-conditional Gaussian distributions; using this model, the test signal $z[n]$ is not Gaussian distributed.

3.3.1 Signal Classification Experiment: Laplacian Multipath

Each coefficient of a multipath filter was drawn independently from a Laplacian random process with parameters $\mathbf{m}_h[n]$, $b[n]$:

$$p(h[n] \mid m_h[n], b[n]) = \frac{1}{2b[n]} \exp\left(-\frac{|h - m_h[n]|}{b[n]}\right),$$

for $n = 0, \ldots, 99$, where we set the $\mathbf{m}_h$ to be $m_h[n] = \delta[n] - 0.6\delta[n-49] + 0.1\delta[n-99]$, and the scale parameter $b[n]$ decays as n grows: $b[n] = 0.2e^{-0.024n}$. The decay parameter coefficients for this experiment were chosen to model oceanic multipath filtering of sonar signals.

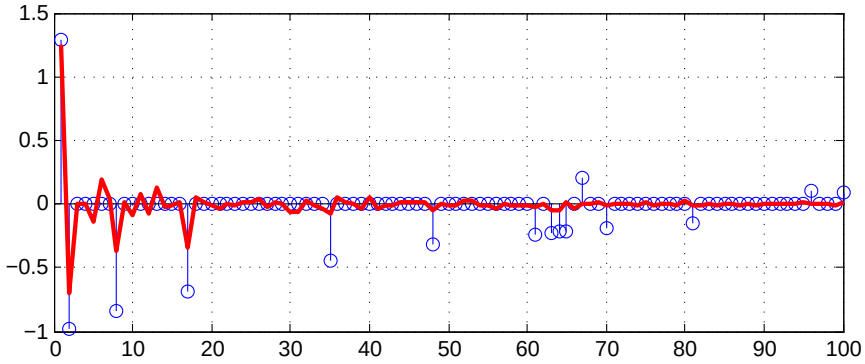


Figure 3.1: Example multipath realization from the k-sparse model (stem), and the deconvolution estimate produced by the joint MAP deconvolution/classifier (solid) at 10 dB SNR.

Test and training signals were drawn i.i.d. from a Gaussian distribution $\mathcal{N}(\mathbf{m}_{x|y}, \Sigma_{x|y})$ where the class y was drawn uniformly between two classes. Two classification scenarios

were considered to test performance: (i) classes that were well-separated by their mean vectors, and (ii) classes whose mean vectors were similar. The mean signals were composed of square and sine waves, and the covariance matrices were Toeplitz with smooth covariance structure. The specific values of $\mathbf{m}_{x|y}$ and $\Sigma_{x|y}$ for each experiment are shown in Table 3.1. Each test signal $\mathbf{z}$ was created by convolving a randomly drawn signal $\mathbf{x}$ with randomly drawn multipath $\mathbf{h}$, and adding a Gaussian white noise realization $\mathbf{w}$ to achieve SNR between -10 and 10 dB, where the SNR is with respect to the multipath signal, $10\log_{10}\frac{\|\mathbf{x}*\mathbf{h}\|^2}{\sigma_w^2}$.

Table 3.1: Simulation parameters for joint MAP / joint QDA experiments. Note that $\text{square}[n] = \text{sgn}(\sin[n])$.

parameter	**class 1**	**class 2**
Well-separated means		
$m_{x\|y}[n]$	$\frac{1}{4}$ square $\left[\frac{6\pi n}{100}\right]$	$\frac{1}{4}$ square $\left[\frac{12\pi n}{100}\right]$
$\Sigma_{x\|y}[m,n]$	$\frac{1}{100}\left(\delta[m-n] + \exp\left(-\frac{\lvert m-n\rvert}{20}\right)\right)$	$\frac{1}{100}\left(\delta[m-n] + \exp\left(-\frac{(m-n)^2}{10}\right)\right)$
Close means		
$m_{x\|y}[n]$	$\frac{1}{4}$ square $\left[\frac{6\pi n}{100}\right]$	$\frac{1}{4}$ sin $\left[\frac{6\pi n}{100}\right]$
$\Sigma_{x\|y}[m,n]$	$\frac{1}{100}\left(\delta\left[m-n\right] + \exp\left(-\frac{\lvert m-n\rvert}{20}\right)\right)$	$\frac{1}{100}\left(\delta[m-n] + \exp\left(-\frac{(m-n)^2}{10}\right)\right)$

The joint QDA classifier was compared to a matched filter that ignores multipath. For the matched filter, the received signal $\mathbf{z}$ is tested against $\mathbf{m}_{x|y}$ for each class. The joint MAP classifier in (3.4) is compared to a matched filter on a blind deconvolution signal estimate. For blind deconvolution, the received signal $\mathbf{z}$ is first denoised by Wiener filtering, then $\hat{\mathbf{h}}$ is estimated using Cabrelli's blind deconvolution method for signals that have undergone unknown multipath filtering [9]. The estimate $\hat{\mathbf{x}}$ is then computed via deconvolution in the Fourier domain. The true signal length was used as a required input to Cabrelli's method. Each of the methods used the true signal and channel statistics, and the true SNR where needed.

3.3.2 Signal Classification Experiment: k-sparse Multipath

The k-sparse experiments are the same as described in the previous subsection, except the multipath filters were generated using a sparse model

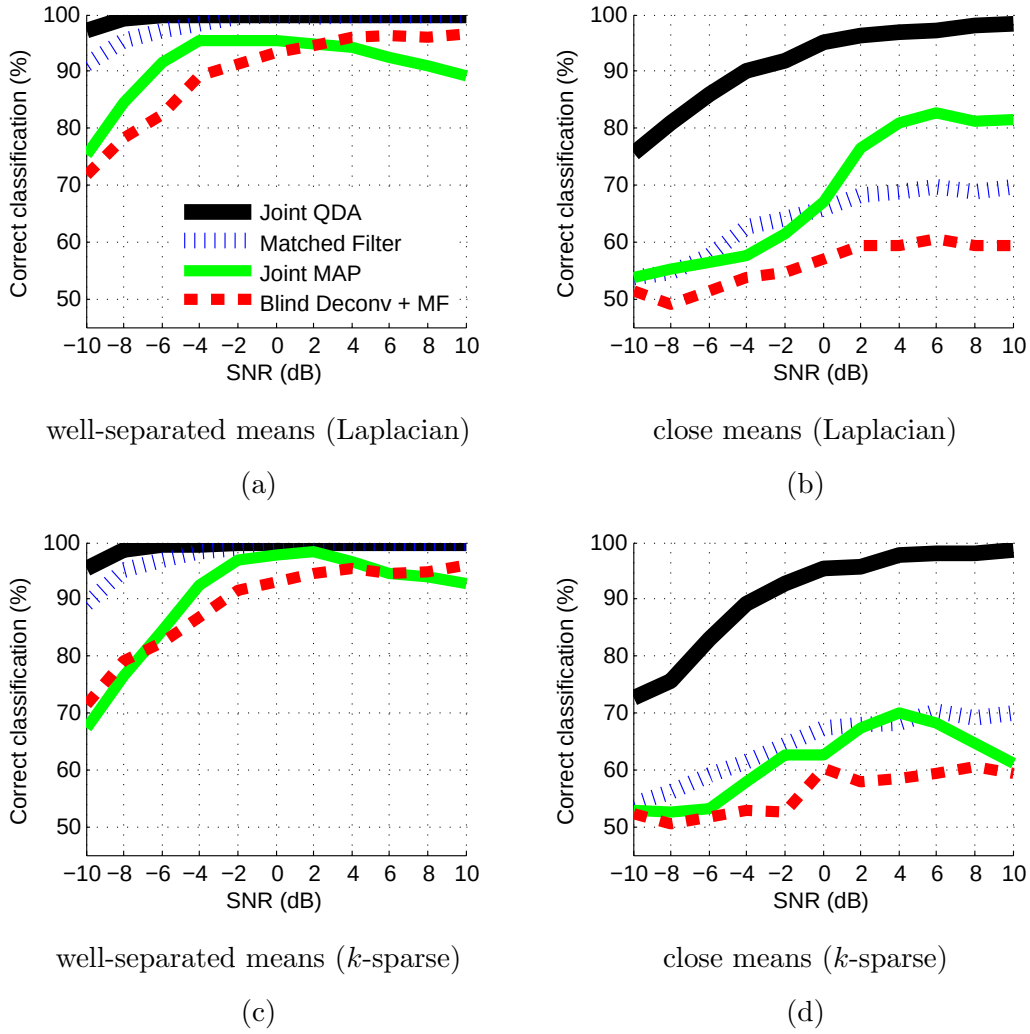
$$h[n] = \sum_{i=1}^{k} \alpha_i \delta[n - d_i],$$

with $k = 15$ nonzero coefficients, delays d_i drawn uniformly on $[0, 99]$, $\alpha_i = \pm e^{-\beta d_i}$ with randomly chosen sign and decay parameter $\beta = 0.0240$ chosen to mimic real underwater acoustic channels. An example realization of a filter h drawn from this model is shown in Fig. 3.1. The diagonal covariance matrix Σ_h is estimated from 1000 samples of the impulse response.

3.3.3 Signal-based Joint QDA and Joint MAP Results

Figure 3.1 shows a reconstructed multipath estimate produced by the joint MAP deconvolution/classifier corresponding to the well-separated means experiment at 10 dB SNR. In this case joint MAP correctly identified the class label. The recovered filter is a reasonable reconstruction of the true filter, but generally underestimates the amplitude of the first coefficients, and does not reliably reconstruct the tail of $\mathbf{h}$. The gross errors can be ascribed to the fact that the optimization problem in (3.4) is not convex, and to the mismatch between the Laplacian prior and k-sparse model.

Classification results in Fig. 3.2 show that the proposed joint QDA classifier dominates the matched filter for both Laplacian multipath in (a) and (b), and for k-sparse multipath in (c) and (d). The means for each class used for (a) and (c) are orthogonal, so the matched filter performs well despite ignoring the multipath. With similar means in (b) and (d), however, the matched filter performs poorly compared to joint QDA. The joint MAP classifier performs well at low SNR, but as predicted, degrades at high SNR. For truly sparse multipath in (c) and (d), the joint MAP approach is unaffected for well-separated means in (c), and mildly affected at high SNR for close means in (d) when compared to results for Laplacian multipath. The ℓ_1 norm is an appropriate heuristic for the k-sparse multipath model, and has been employed elsewhere to recover sparse solutions [10].



well-separated means (Laplacian)

(a)

close means (Laplacian)

(b)

well-separated means (k-sparse)

(c)

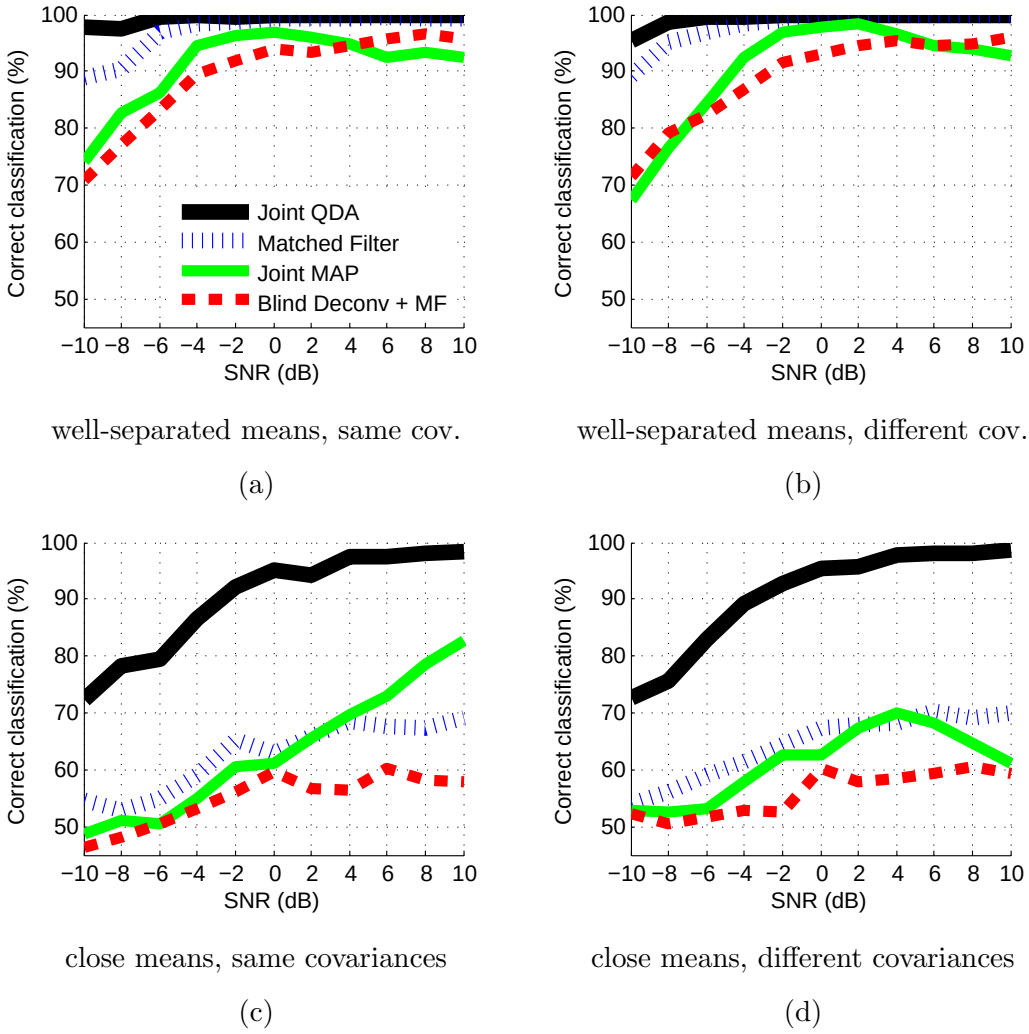
close means (k-sparse)

(d)

Figure 3.2: Classification accuracy for four experiments using multipath generated from a Laplacian model in (a) and (b), and a k-sparse model in (c) and (d). The results are averaged over 1000 i.i.d. test signals for each SNR point.

3.4 Conclusions

Classification methods were proposed that jointly consider the effects of multipath distortion with classification. In particular, a joint MAP deconvolution/classifier was derived that incorporates first and order statistics of the channel and yields a MAP solution for the recovered signal $\hat{\mathbf{x}}$, the recovered filter $\hat{\mathbf{h}}$ and the class label y^*. Two drawbacks of the joint



well-separated means, same cov. (a)

well-separated means, different cov. (b)

close means, same covariances (c)

close means, different covariances (d)

Figure 3.3: Classification accuracy for four experiments using multipath generated from a sparse model. The results are averaged over 500 i.i.d. test signals for each SNR.

MAP algorithm is that it is not convex, and that it theoretically performs poorly at high SNR. The first problem might be addressed by maximizing the marginal $p(\mathbf{h}, y|\mathbf{z})$ which yields a convex expression, but requires more complicated optimization approaches (e.g., an EM algorithm similar to Goodman and Lam [42]). The second problem arises since regularization scales with the noise power σ_w^2, which may be replaced by a fixed penalty that can be chosen via cross-validation.

It was hypothesized that better classification performance could be gained by marginalizing over $\mathbf{x}$ and $\mathbf{h}$. To that end, a joint QDA classifier was presented that accounts for the LTI corruption probabilistically. Experiments showed that the joint QDA classifier outperformed the joint MAP classifier and a classifier based on blind deconvolution.

The joint MAP and joint QDA classifiers presented in this chapter utilize discrete-time signals as feature vectors. In practice, the length of the signals of interest may be very large, so that estimating covariance matrices may be ill-posed and inverting the matrices may be computationally intractable. Therefore, in the next chapter, a feature-based joint QDA classifier is developed.

Chapter 4

JOINT QDA FOR SUBBAND ENERGY FEATURES

> It's less of a sin to be simple and wrong than to be complicated and wrong.
>
> Dr. Henry Eyring, chemist

Signal-based deconvolution and classification methods—including blind deconvolution, joint MAP deconvolution and classification and the joint QDA classifier—are computationally prohibitive for signals captured at high sample rates. For L-length sampled signals, Cabrelli's blind deconvolution method requires the inversion of an $L \times L$ Toeplitz matrix, which can be solved in $O(L \log L)$ operations. The joint MAP deconvolution and classification requires more iterations to converge as L increases, and requires (as does joint QDA) inversion of an $L \times L$ covariance matrix, which in general is complexity $O(L^3)$. To decrease the computational burden and possibly increase classification performance, an alternative is to classify based on features that represent important characteristics of the signals and provide good class discrimination [17]. The hope is that classes can be well-discriminated by features of significantly smaller dimensionality $d \ll L$.

Unless channel-invariant features are used, a classifier trained on features of $\{x_i[n]\}_{i=1}^N$ will not generally be applicable to classify the features of $z(t)$ directly. However, if a functional relationship can be found that relates the training samples $\{\mathbf{x}_i\}_{i=1}^N$ to the test sample $\mathbf{z}$, then a suitable classifier may be constructed. In this chapter, the functional relationship between subband energies of $x[n]$ and $z[n]$ is used to derive a feature-based joint QDA classifier. Subband energy features are a useful and frequently utilized feature choice in many signal processing applications. The joint QDA classifier is compared to other classifiers in simulated underwater passive acoustic experiments. Portions of this chapter are published in [2].

4.1 Joint QDA Using Second Order Statistics of Subband Energy

It is assumed that to classify a subband energy feature vector $\mathbf{u}_z$, training pairs $\{(\mathbf{u}_{x_i}, y_i)\}_{i=1}^N$, auxiliary channel samples $\{\mathbf{u}_{h_i}\}_{i=1}^M$, and the noise power are provided; however, the phase of the training signals (e.g., provided by $\mathbf{x}_i^f$) and channel impulse responses is not assumed to be provided. It will be shown that a robust classifier can be derived without the need to model the phase of the signal, channel or noise.

Model $\mathbf{H}^f$, $\mathbf{X}^f$ and $\mathbf{W}^f$ as mutually independent random vectors, and $\mathbf{U}_x = \mathbf{X}^f \cdot \mathbf{X}^{f*}$ so that

$$\mathbf{U}_z = \mathbf{U}_h \cdot \mathbf{U}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}. \tag{4.1}$$

We assume that $\mathbf{W}^f \in \mathbb{C}^d$ is a proper complex Gaussian random vector with $\mathrm{E}\left[\mathbf{W}^f\right] = 0$ and $\mathrm{Cov}\left[\mathbf{W}^f\right] = \sigma_w^2 I$.

To derive the joint QDA classifier for subband energy features we need only compute $\bar{\mathbf{U}}_{z|y} = \mathrm{E}\left[\mathbf{U}_z|\, y\right]$ and $\Sigma_{u_{z|y}} = \mathrm{Cov}\left[\mathbf{U}_z|\, y\right]$. The mean is given by

$$\begin{aligned}
\mathrm{E}\left[\mathbf{U}_z|y\right] &\overset{a}{=} \mathrm{E}\left[\mathbf{U}_h \cdot \mathbf{U}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\} |\, y\right] \\
&\overset{b}{=} \mathrm{E}\left[\mathbf{U}_h\right] \cdot \mathrm{E}\left[\mathbf{U}_x|\, y\right] + \mathrm{E}\left[\mathbf{U}_w\right] + 2\,\mathrm{Re}\left\{\mathrm{E}\left[\mathbf{X}^f|\, y\right] \cdot \mathrm{E}\left[\mathbf{H}^f\right] \cdot \mathrm{E}\left[\mathbf{W}^f\right]^*\right\} \\
&\overset{c}{=} \bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_{x|y} + \sigma_w^2 I,
\end{aligned} \tag{4.2}$$

where the independence assumptions and the fact that $\mathrm{Re}\left\{\mathrm{E}\left[\cdot\right]\right\} = \mathrm{E}\left[\mathrm{Re}\left\{\cdot\right\}\right]$ have been used to reduce (a) to (b); and (b) reduces to (c) since $\mathrm{E}\left[\mathbf{W}^f\right] = 0$.

The covariance $\Sigma_{u_{z|y}}$ is derived from Equation (B.23) in the appendix by conditioning on the class y:

$$\Sigma_{u_{z|y}} = \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_{x|y}} + \Sigma_{u_h} \cdot \bar{\mathbf{U}}_{x|y} \bar{\mathbf{U}}_{x|y}^T + \sigma_w^4 I + 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_{x|y}\right). \tag{4.3}$$

Using (4.2) and (4.3), the joint QDA classifier for subband energy features is given by

$$y^* = \arg\max_y \left(\mathbf{u}_z - \bar{\mathbf{U}}_{z|y}\right)^T \Sigma_{u_{z|y}}^{-1} \left(\mathbf{u}_z - \bar{\mathbf{U}}_{z|y}\right) + \log\left|\Sigma_{u_{z|y}}\right|.$$

It is assumed that $\bar{\mathbf{U}}_{x|y}$ and $\Sigma_{u_{x|y}}$ can be estimated from the training pairs $\{(\mathbf{u}_{x_i}, y_i)\}_{i=1}^N$, that $\bar{\mathbf{U}}_h$ and Σ_{u_h} can be estimated from the auxiliary channel data $\{\mathbf{u}_{h_i}\}_{i=1}^M$, and that σ_w^2 is known.

4.1.1 Modeling Subband Energies $\mathbf{U}_z$ as Gaussian

The vector of subband energies $\mathbf{u}_z$ is in fact non-negative, but modeling $\mathbf{U}_z$ as Gaussian permits $\mathbf{u}_z < 0$ with nonzero probability. The Gaussian assumption is motivated by computational convenience and by the fact that the Gaussian distribution is the maximum entropy distribution over $\mathbb{R}^d$ given only $\mathrm{E}[\mathbf{U}_z|y]$ and $\mathrm{Cov}[\mathbf{U}_z|y]$. It may be more suitable to relax the Gaussian assumption, and instead consider the maximum entropy distribution over the positive orthant $\mathbb{R}^d_+$. However, the maximum entropy distribution over $\mathbb{R}^d_+$ is the multivariate truncated normal distribution which requires cumbersome multi-dimensional lookup tables of the cumulative distribution function [37]. Similarly, using a multivariate Rayleigh model is analytically cumbersome and computationally challenging.

In practice, the Gaussian model may not be in gross violation of the constraint $\mathbf{u}_z > 0$. Commonly, and as in the experiments in this chapter, a frequency bin k is selected so that the features $u_{x_i}[k] \gg 0$ and exhibit low variance for a given class. The resulting Gaussian model is concentrated around the features in the positive orthant, and the area under the distribution's tail for $\mathbf{u}_z < 0$ is quite small. However, in general, the assumption that $\mathbf{U}_z$ is Gaussian admits model bias into the joint QDA classifier.

4.2 Local Joint QDA

In this section, the model bias of joint QDA is reduced by relaxing the assumption that $p(\mathbf{u}_z|y)$ is globally Gaussian. A standard approach to model an arbitrary distribution is the Gaussian mixture model (GMM). By properly choosing the appropriate number of Gaussian components, GMMs can be very flexible [30], but can have high estimation variance due to local minima of the expectation-maximization algorithm required to learn the parameters of the GMM. Instead, another approach is used that has recently been shown to work well, which is to apply the Gaussian model locally to the nearest-neighbors of the test sample, an approach aptly termed *local QDA* [23].

A local joint QDA classifier is proposed in which, given an observation $\mathbf{z}$, the Gaussian distribution is fitted to only the training pairs that correspond to the expected nearest neighbors *for each class*. Expected nearest neighbors are defined as follows.

Definition 1. *Expected Nearest Neighbor.* Model random training vector $\mathbf{Z}_i$ as Gaussian with mean $\bar{\mathbf{Z}}_i$ and covariance Σ_{z_i}. Given a test sample $\mathbf{z}$, the expected nearest neighbor of $\mathbf{z}$ is the random vector $\mathbf{Z}_\ell$, where

$$\begin{aligned}\ell &\stackrel{\triangle}{=} \arg\min_i \ \mathrm{E}\left[\|\mathbf{z}-\mathbf{Z}_i\|^2\right] \\ &= \arg\min_i \ \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\,\mathrm{E}\left[\mathbf{Z}_i\right] + \mathrm{E}\left[\mathbf{Z}_i^T\mathbf{Z}_i\right] \\ &= \arg\min_i \ \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\bar{\mathbf{Z}}_i + \mathrm{tr}\,\Sigma_{z_i} + \bar{\mathbf{Z}}_i^T\bar{\mathbf{Z}}_i \\ &= \arg\min_i \ \|\mathbf{z}-\bar{\mathbf{Z}}_i\|^2 + \mathrm{tr}\,\Sigma_{z_i}. \end{aligned} \tag{4.4}$$

Note that the nearest neighbor in Definition 1 depends on both $\bar{\mathbf{Z}}_i$ and Σ_{z_i} of a random training vector $\mathbf{Z}_i$. The second nearest neighbor is found in similar fashion, after $\mathbf{Z}_\ell$ has been excluded from the set of candidate neighbors, and so on for the subsequent nearest neighbors.

Let $\mathcal{X}_y = \{\mathbf{x}_i : y_i = y\}$ for each class y. Given observation $\mathbf{z}$, let $\mathcal{K}_y$ be the set of training samples in $\mathcal{X}_y$ that correspond to the k_y expected nearest neighbors of $\mathbf{z}$.

For discrete-time features $\mathbf{Z}_\ell = \mathbf{H} * \mathbf{x}_\ell + \mathbf{W}$ is the expected nearest neighbor to $\mathbf{z}$ by Definition 1 with $\bar{\mathbf{Z}}_i = \bar{\mathbf{H}} * \mathbf{x}_i$ and $\Sigma_{z_i} = \Sigma_h * *\mathbf{x}_i\mathbf{x}_i^T + \sigma_w^2$. Then, the mean and covariance of Gaussian likelihood $p(\mathbf{z}|y)$ is calculated as the sample mean and covariance, respectively, of $\mathcal{X}_y$. The local class-conditional distribution $p(\mathbf{z}|y) = \mathcal{N}(\mathbf{z}; \bar{\mathbf{Z}}_{k_y}, \Sigma_{zk_y})$ where

$$\bar{\mathbf{Z}}_{k_y} = \bar{\mathbf{H}} * \bar{\mathbf{X}}_{k_y}, \qquad\qquad \Sigma_{zk_y} = \Sigma_h * *\Sigma_{xk_y} + \sigma_w^2 I,$$

where $\bar{\mathbf{X}}_{k_y}$ and Σ_{xk_y} are taken as the sample mean and covariance of the k_y elements of $\mathcal{K}_y$.

For subband energy features, Definition 1 is applied for a test signal $\mathbf{u}_z$ and random variable $\mathbf{U}_{z_i} = \mathbf{U}_h \cdot \mathbf{u}_{x_i} + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{H}^f \cdot \mathbf{x}_i^f \cdot \mathbf{W}^{f*}\right\}$ which has mean and covariance

$$\begin{aligned}\bar{\mathbf{U}}_{z_i} &= \bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i} + \sigma_w^2 \\ \Sigma_{u_{z_i}} &= \Sigma_{u_h} \cdot \mathbf{u}_{x_i}\mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right).\end{aligned}$$

The covariance $\Sigma_{u_{z_i}}$ is derived from (B.23) in the appendix by substituting $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$ and

$\Sigma_{u_x} = 0$. Then, the distribution $p(\mathbf{u}_z|y) = \mathcal{N}(\mathbf{u}_z; \bar{\mathbf{U}}_{zk_y}, \Sigma_{u_z k_y})$ is specified by parameters

$$\bar{\mathbf{U}}_{zk_y} = \bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_{xk_y} + \sigma_w^2 \mathbf{1},$$
$$\Sigma_{u_z k_y} = \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_x k_y} + \Sigma_{u_h} \cdot \bar{\mathbf{U}}_{xk_y} \bar{\mathbf{U}}_{xk_y}^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_{xk_y}\right),$$

where $\bar{\mathbf{U}}_{xk_y}$ and $\Sigma_{u_x k_y}$ are estimated from the k_y elements of $\mathcal{K}_y$, and $\bar{\mathbf{U}}_h$ and Σ_{u_h} are estimated from auxiliary set $\{\mathbf{u}_{h_i}\}_{i=1}^M$.

The neighborhood size k_y for each class y is a parameter of local joint QDA that must be chosen based on prior knowledge or via crossvalidation. Local joint QDA generalized joint QDA since one may choose $k_y = |\mathcal{X}_y|$. Experiments to test local joint QDA are presented in Chapter 6.

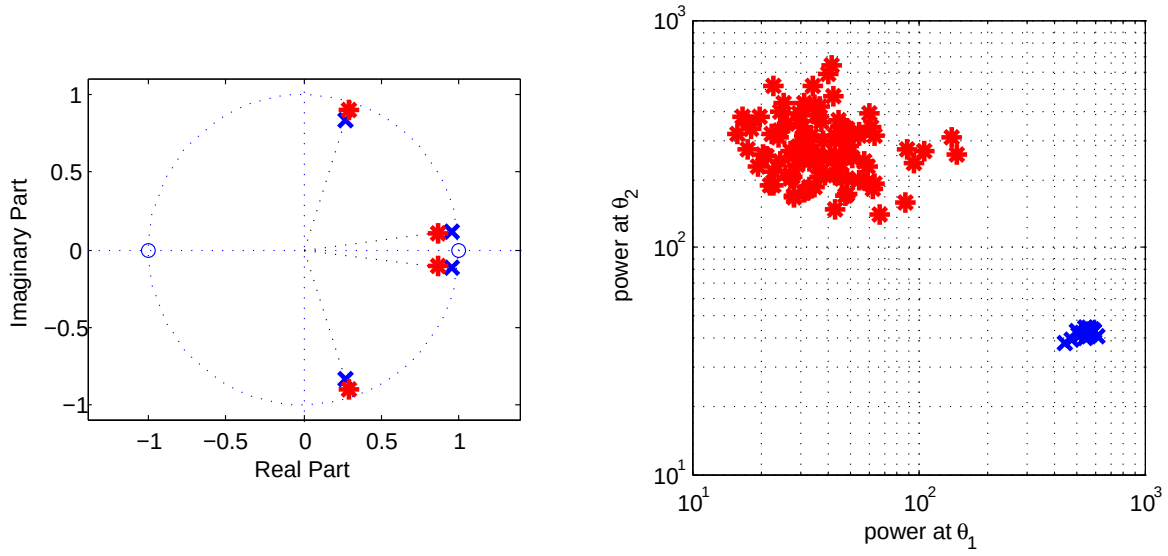
4.3 Experiments: Feature-based Classification of Simulated Signals

The proposed joint QDA using subband energy features is compared to two alternate classification approaches that use subband energy features. First, a "deconvolution" approach removes the mean effect of the channel using $\mathrm{E}[\mathbf{U}_h]$ and the noise energy σ_w^2 to form an estimate $\hat{\mathbf{u}}_x = \frac{[\mathbf{u}_z - \sigma_w^2]}{[\bar{\mathbf{U}}_h]}$, where $\frac{[\mathbf{a}]}{[\mathbf{b}]}$ denotes Hadamard (elementwise) division of vectors $\mathbf{a}$ and $\mathbf{b}$. The estimate $\hat{\mathbf{u}}_x$ is then compared to training pairs $\{(\mathbf{u}_{x_i}, y_i)\}_{i=1}^N$ using a standard classifier (e.g., QDA, SVM, or k-NN). This approach utilizes the sample mean (but not the sample covariance) of the auxiliary channel samples.

Second, a "normalization" classifier approach ignores the auxiliary samples $\{\mathbf{u}_{h_i}\}_{i=1}^M$ altogether. Training signals $\{x[n]_i\}_{i=1}^N$ are first normalized by their total energy before extracting training samples $\{\hat{\mathbf{u}}_{x_i}\}_{i=1}^N$. A test vector $\hat{\mathbf{u}}_z$ is also formed by first normalizing the signal $z[n]$ by its total energy. This approach corrects for the gross attenuation of the test signal caused by the channel, but not the spectral shaping.

Joint QDA is compared to both the normalization and deconvolution approaches for QDA, 1-NN, and a support vector machine (SVM) [30].

The task of classifying narrow-band signals is considered, where the signals are corrupted by unknown multipath due to propagation in a shallow ocean channel. To simulate narrow-



(a) (b)

Figure 4.1: (a) Pole-zero plot showing the mean location of the poles for class 1 ($\times$) and class 2 ($*$) for the `easy` case, and (b) scatter-plot of the classes in log-feature space.

band signals, training and test signals are generated i.i.d. using the z-transform model

$$X_y[z] = \frac{(z-1)(z+1)}{(z-p_{y,1})(z-p_{y,1}^*)(z-p_{y,2})(z-p_{y,2}^*)} \quad \text{for class } y = 1, 2,$$

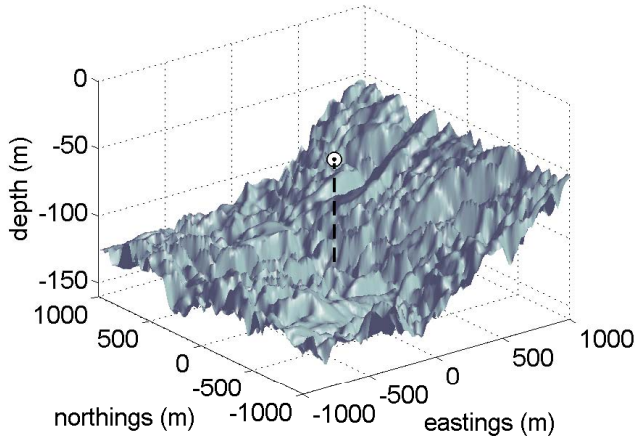
where $X[z]$ denotes the z-transform of the discrete-time signal $x[n]$. The location of each class-conditional pole $p_{y,\ell}, \ell \in \{1, 2\}$ is drawn randomly from the model $a_{y,\ell} \exp(j\theta_\ell)$, where θ_ℓ is fixed, and $\mathbf{a}_y = [a_{y,1},\ a_{y,2}]^T$ is multivariate Gaussian distributed with mean $\mathbf{m}_{a|y}$ and covariance matrix $\Sigma_{a|y}$. Although the vector $\mathbf{a}_y$ for each class y is Gaussian distributed, the signals in feature space are not. Three instances of the experiment are considered for different choices of $\mathbf{m}_{a|y}$ that result in different class separation: `easy`, `medium` and `hard`. The parameters $\mathbf{m}_{a|y}$ and $\Sigma_{a|y}$ for each instance of the experiment are shown in Table 4.1, and we set $\theta_1 = \frac{1}{50}$ and $\theta_2 = \frac{1}{5}$. Figure 4.1 shows an example pole-zero plot and corresponding log-feature space scatterplot for a well-separated (`easy`) case. Note that since all poles and zeros lie within the unit circle, for each case the selected parameters correspond to a realization of a minimum phase signal, which could be produced from natural sources.

Table 4.1: Pole Magnitude Distribution for Feature-based Classification Experiments

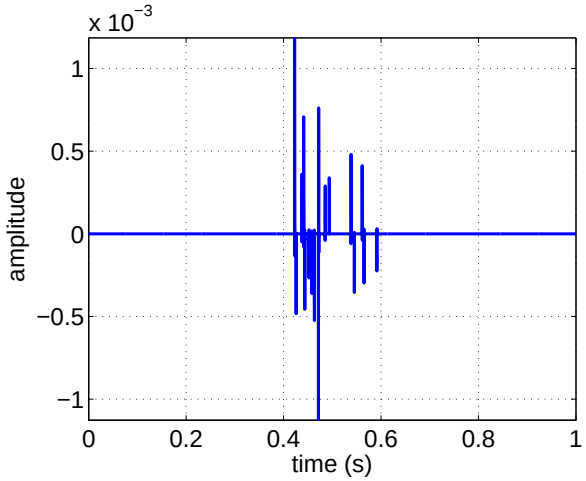
Parameter	**Class 1**	**Class 2**
$\Sigma_{a\|y}$	$\begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 9.00 \end{bmatrix} \times 10^{-6}$	$\begin{bmatrix} 6.00 & -0.80 \\ -0.80 & 1.00 \end{bmatrix} \times 10^{-4}$
	`hard`	
$\mathbf{m}_{a\|y}$	$\begin{bmatrix} 0.945 & 0.905 \end{bmatrix}^T$	$\begin{bmatrix} 0.909 & 0.948 \end{bmatrix}^T$
	`medium`	
$\mathbf{m}_{a\|y}$	$\begin{bmatrix} 0.945 & 0.875 \end{bmatrix}^T$	$\begin{bmatrix} 0.879 & 0.948 \end{bmatrix}^T$
	`easy`	
$\mathbf{m}_{a\|y}$	$\begin{bmatrix} 0.965 & 0.875 \end{bmatrix}^T$	$\begin{bmatrix} 0.875 & 0.948 \end{bmatrix}^T$

Test and training signals were generated by taking i.i.d. draws of poles as described above, and taking 5000 evenly-spaced samples around the unit circle in the z-transform domain, so that the length of each signal corresponds to 1.25 seconds, sampled at 4 kHz. The subband energy at frequencies θ_1 and θ_2 are extracted from each signal and used as classification features. The parameters in Table 4.1 were chosen such that the generated test and training signals were linearly separable in the subband energy feature space.

Channel impulse responses were drawn i.i.d. in the following manner. A receiver is placed at a depth of 50m in a simulated shallow water channel, as shown in Fig. 4.2. Source locations were drawn uniformly from the cube 2 km across north and east and 150 m deep; locations falling below the ocean floor are discarded and redrawn. Channel impulse responses were generated by propagating an impulsive source from the random source locations to the receiver using the *CASS Eignenray* routine provided in the Sonar Simulation Toolset [25]. Impulse responses were sampled at 4 kHz. The ocean environment is set up to be fairly extreme, but static. A nominal sound speed profile was imposed, and modeled the ocean bottom to contain sandy gravel with mean grain size 2mm. Surface roughness is governed by the wind speed, which is set to 15 km/hr. The channel geometry



(a)



(b)

Figure 4.2: (a) Simulated ocean bathymetry with a single receiver (marked by $\odot$) at $(0, 0, -50)$ m, and (b) a sample channel impulse response for a source located at $(460, 250, -70)$ m, generated by the Sonar Simulation Toolset [25].

and a sample channel impulse response are shown in Fig. 4.2.

Maximum likelihood estimates of $\bar{\mathbf{U}}_{x|y}$ and $\Sigma_{u_{x|y}}$ were computed from N=1000 training feature vectors, and maximum likelihood estimates of $\bar{\mathbf{U}}_h$ and Σ_{u_h} were computed from M=1000 auxiliary channel samples. The test signals were corrupted with randomly drawn multipath, and then i.i.d. white noise was added so that the multipath-corrupted-signal to

noise ratio was varied between -10 and 10 dB. Classification results were averaged over 10000 trials for each SNR.

The proposed joint QDA classifier was compared with $k_y = |\mathcal{X}_y|$ (experiments with *local* joint QDA will be given in Chapter 6) to traditional QDA, support vector machine (SVM) with a linear kernel, and 1-NN classifiers using both the "deconvolution" and "normalization" approaches described in Section 4.3. Although the 1-NN, QDA and SVM classifiers produce a non-linear, quadratic, and linear decision boundary, respectively, in each of the three simulations, the training and test data were linearly separable. Using this as prior information, the regularization term C in the linear SVM was set to a large value, $C = 10^7$ [20]. With this setup, none of the classifiers require cross-validation.

4.3.1 Simulation Results

Results for each experiment are shown in Fig. 4.3. Joint QDA performs markedly better than the other approaches when the classes are difficult to separate. The normalization QDA performs poorly in all datasets, and deconvolution QDA only exhibits utility for high SNR when classes are well separated. In general, the deconvolution methods outperformed the normalization methods, which comes at no surprise, since the deconvolution methods actually utilize the auxiliary features $\{\mathbf{u}_{h_i}\}_{i=1}^{M}$. As class separation increases, the performance of deconvolution SVM and deconvolution 1-NN is similar to joint QDA.

4.4 Conclusions

A joint QDA classifier was derived for subband energie features. In underwater acoustic simulations involving narrowband data, the joint QDA classifier compared favorably to QDA, SVM and 1-NN classifiers that employ deconvolution and normalization.

The SVM and 1-NN classification methods presented in this chapter represent a naive approach to utilizing the auxiliary training samples. In the case of deconvolution SVM and deconvolution 1-NN, only the sample mean of $\{\mathbf{u}_{h_i}\}_{i=1}^{M}$ is used. In the case of normalization SVM and normalization 1-NN, the auxiliary training samples are discarded in favor of simple energy normalization. The following chapter presents kernels for SVM classifiers that better incorporate the auxiliary training samples $\{\mathbf{u}_{h_i}\}_{i=1}^{M}$.

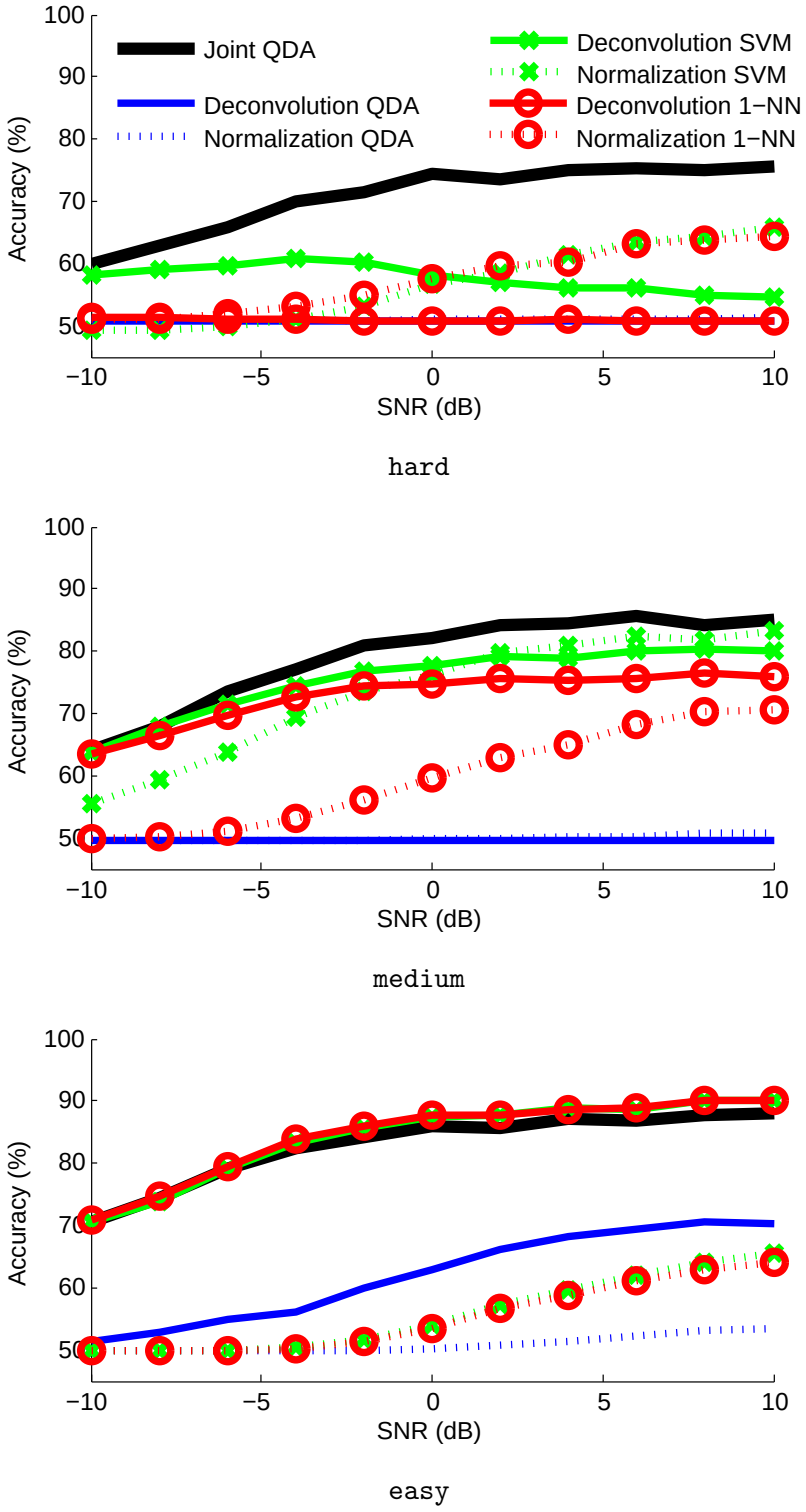


Figure 4.3: Results for feature-based classification on simulated data.

Chapter 5

CHANNEL-ROBUST KERNELS FOR SUPPORT VECTOR MACHINES

> To develop a suitable model, I ask myself, "How would I act if I were an atom or molecule and found myself in this situation?"
>
> Dr. Henry Eyring, chemist

Built into the joint QDA classifier is the convolution relationship between the unknown signal $x[n]$ and the true signal $z[n]$ via the class-conditional distribution $p(\mathbf{z}|y)$. In this chapter, the convolution relationship is also built into a support vector machine (SVM) by constructing appropriate kernel functions. Although the focus is on SVMs, the resulting kernels are applicable to any kernel method.

As reviewed in Chapter 2, methods have been proposed for learning invariant classifiers. One flexible approach—the method of virtual examples (VEs)—can be used to train any classifier by creating a large artificial training set from $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ and $\{\mathbf{h}_i\}_{i=1}^{M}$. However, training an SVM using the VE method has a complexity of $O(M^3 \times N^3)$. Rather than increase the dataset by a factor of M, two alternative approaches are introduced—the *expected kernel* and the *projected RBF kernel*—that incorporate the stochastic channel into the kernel definition. For both approaches, the ith training sample $\mathbf{x}_i$ is mapped to a probability distribution $p_{Z_i|x_i}$ over the domain of noisy channel-corrupted signals. Then a kernel is defined that acts on two probability distributions in the noisy domain. The two approaches differ in how the samples are mapped to probability distributions, and (relatedly) how the kernels are defined. Closed-form solutions are derived for the proposed kernels for discrete-time features, images, and for subband energy features, summarized in Tables 5.1, 5.2 and 5.3, respectively.

Table 5.1: Expected and Projected RBF Kernels for Randomly Filtered Discrete-time Signals

	Training Kernel	Test Kernel
Expected RBF	$\mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, \Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right)$
Clean-train Expected RBF	$\mathcal{N}\left(\bar{\mathbf{x}}_i; \bar{\mathbf{x}}_j, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right)$
Projected RBF	$\mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, R_{z_i} + R_{z_j}\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, R_{z_i} + \tilde{R}_z\right)$
Clean-train Projected RBF	$\mathcal{N}\left(\bar{\mathbf{x}}_i; \bar{\mathbf{x}}_j, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, R_{z_i} + \tilde{R}_z\right)$

where,

$$\bar{\mathbf{Z}}_i = \mathbf{x}_i * \bar{\mathbf{H}} \text{ and } \Sigma_{z_i} = \Sigma_h ** \mathbf{x}_i \mathbf{x}_i^T + \sigma_w^2 I$$
$$R_{z_i} = \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}} \bar{\mathbf{H}}^T\right) + \Sigma_{z_i}$$
$$\tilde{R}_z = \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}} \bar{\mathbf{H}}^T\right) + \Sigma_h ** \hat{\mathbf{x}} \hat{\mathbf{x}}^T + \sigma_w^2 I$$
$$\mathbf{z} = \bar{\mathbf{H}} * \hat{\mathbf{x}}$$

Table 5.2: Expected and Projected RBF Kernels for Randomly Filtered Images

	Training Kernel	Test Kernel
Expected RBF	$\mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, \Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right)$
Clean-train Expected RBF	$\mathcal{N}\left(\bar{\mathbf{x}}_i; \bar{\mathbf{x}}_j, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right)$
Projected RBF	$\mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, R_{z_i} + R_{z_j}\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, R_{z_i} + \tilde{R}_z\right)$
Clean-train Projected RBF	$\mathcal{N}\left(\bar{\mathbf{x}}_i; \bar{\mathbf{x}}_j, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, R_{z_i} + \tilde{R}_z\right)$

where,

$$\text{fold}(\bar{\mathbf{Z}}_i) = \text{fold}\left(\mathbf{x}_i\right) ** \text{fold}\left(\bar{\mathbf{H}}\right)$$

$$\text{fold}(\Sigma_{z_i}) = \text{fold}\left(\Sigma_h\right) **** \left(\text{fold}\left(\mathbf{x}_i\right) \circ \text{fold}\left(\mathbf{x}_i\right)\right) + \sigma_w^2 I$$

$$R_{z_i} = C_h + \Sigma_{z_i}$$

$$\text{fold}(\tilde{R}_z) = \text{fold}(C_h) + \text{fold}\left(\Sigma_h\right) **** \left(\text{fold}\left(\hat{\mathbf{x}}\right) \circ \text{fold}\left(\hat{\mathbf{x}}\right)\right) + \sigma_w^2 I$$

$$\text{fold}(C_h) = \text{fold}\left(\frac{\gamma^{-1}}{2} I\right) **** \left(\text{fold}\left(\Sigma_h\right) + \text{fold}\left(\bar{\mathbf{H}}\right) \circ \text{fold}\left(\bar{\mathbf{H}}\right)\right)$$

$$\text{fold}(\mathbf{z}) = \text{fold}\left(\bar{\mathbf{H}}\right) ** \text{fold}\left(\hat{\mathbf{x}}\right)$$

Table 5.3: Expected and Projected RBF Kernels for Subband Energies of Randomly Filtered Signals

	Training Kernel	Test Kernel
Expected RBF	$\mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$
Clean-train Expected RBF	$\mathcal{N}\left(\bar{\mathbf{u}}_{x_i}; \bar{\mathbf{u}}_{x_j}, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$
Projected RBF	$\mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, R_{u_{z_i}} + R_{u_{z_j}}\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right)$
Clean-train Projected RBF	$\mathcal{N}\left(\bar{\mathbf{u}}_{x_i}; \bar{\mathbf{u}}_{x_j}, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right)$

where,

$$\bar{\mathbf{U}}_{z_i} = \mathbf{u}_{x_i} \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1} \text{ and } \Sigma_{u_{z_i}} = \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right)$$

$$R_{u_{z_i}} = \frac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_{z_i}}$$

$$\tilde{R}_{u_z} = \frac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_h} \cdot \hat{\mathbf{u}}_x \hat{\mathbf{u}}_x^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\mathbf{u}_z - \sigma_w^2 \mathbf{1}\right)$$

$$\hat{\mathbf{u}}_x = \frac{\left[\mathbf{u}_z - \sigma_w^2 \mathbf{1}\right]}{\left[\bar{\mathbf{U}}_h\right]}$$

5.1 Expected Kernels

A preliminary version of the expected kernel was presented at a recent conference [34]. Consider the random signal resulting from propagating the features $\mathbf{x}_i$ of the ith training signal through a random noisy channel, and let the feature vector computed from that random signal be the random feature vector $\mathbf{Z}_i \sim p_{Z_i|x_i}$. Then the channel can be taken into account by training an SVM with a kernel that acts on the random feature vectors $\{\mathbf{Z}_i\}_{i=1}^N$ corresponding to the training signals. To that end, given any kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, define the expected kernel $K_{\exp}$ to be the following function of two distributions:

$$\begin{aligned} K_{\exp}(p_{Z_i}, p_{Z_j}) &\stackrel{\triangle}{=} \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j|\mathbf{x}_i,\mathbf{x}_j}\left[K\left(\mathbf{Z}_i, \mathbf{Z}_j\right)\right], \\ &= \iint p_{Z_i|x_i}(\mathbf{z}_i) p_{Z_j|x_j}(\mathbf{z}_j) K(\mathbf{z}_i, \mathbf{z}_j) d\mathbf{z}_i d\mathbf{z}_j. \end{aligned} \tag{5.1}$$

The expected training kernel can be interpreted as averaging the similarity of all possible channel corruptions of $\mathbf{x}_i$ and $\mathbf{x}_j$ weighted by their probability density. To compute the kernel between a training sample and a test sample, let the probability distribution of the test sample be a Dirac delta distribution with all of its support on the feature vector $\mathbf{z}$ computed from the test signal $z[n]$, that is, $p_Z(\mathbf{z}') = \delta(\mathbf{z}' - \mathbf{z})$. The expected kernel given in (5.1) is a legitimate kernel because it is an inner product between its two inputs, where the inner product is weighted by the positive definite function $K(\cdot, \cdot)$, analogous to a discrete inner product of the form $< a, b >_K = a^T K b$ for some positive definite matrix K.

Since the distribution $p_{Z_i|x_i}$ is a function of the training point $\mathbf{x}_i$, for notational simplicity, we write $K_{\exp}(\mathbf{x}_i, \mathbf{x}_j)$ for (5.1), and $K_{\exp}(\mathbf{z}, \mathbf{x}_i)$ for the corresponding kernel between $\delta(\mathbf{z}' - \mathbf{z})$ and $p_{Z_i|x_i}$.

5.1.1 Expected Kernel SVM Compared to Virtual Examples SVM

The expected kernel used with an SVM results in a different objective than the VE method used with an SVM. Let $\{(\mathbf{z}_{ij}, y_{ij})\}_{i,j=1}^{N,M}$ be virtual examples training pairs generated from the powerset of $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and $\{\mathbf{h}_j\}_{j=1}^M$. Let $L(f(\mathbf{x}), y) = (1 - yf(\mathbf{x}))_+$ be the hinge loss, and λ be a regularization parameter. In the noiseless case (so that $\mathrm{E}_{\mathbf{H}}[\cdot] = \mathrm{E}_{\mathbf{Z}|\mathbf{x}}[\cdot]$), and ignoring the bias b, take the limit of the SVM objective function in Equation (1.6) as the

number of auxiliary channel features M increases:

$$\lim_{M\to\infty} \underset{\{\alpha_{ij}\}}{\arg\min}\ \frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M} L\left(\sum_{j=1}^{N}\sum_{m'=1}^{M}\alpha_{jm'}y_{jm'}K(\mathbf{z}_{im},\mathbf{z}_{jm'}),y_i\right) + \lambda\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{j=1}^{N}\sum_{m'=1}^{M}\alpha_{im}y_{im}\alpha_{jm'}y_{jm'}K(\mathbf{z}_{im},\mathbf{z}_{jm'}),$$

which converges in probability (by the law of large numbers) to

$$\xrightarrow{p} \lim_{M\to\infty} \underset{\{\alpha_{ij}\}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{\mathbf{Z}_i|\mathbf{x}_i}\left[L\left(\sum_{j=1}^{N}\sum_{m'=1}^{M}\alpha_{jm'}y_{jm'}K(\mathbf{Z}_i,\mathbf{z}_{jm'}),y_i\right)\right] + \lambda\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{j=1}^{N}\sum_{m'=1}^{M}\alpha_{im}y_{im}\alpha_{jm'}y_{jm'}K(\mathbf{z}_{im},\mathbf{z}_{jm'}).$$

However, the expected kernel SVM solves the objective function

$$\underset{\{\alpha_i\}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N} L\left(\sum_{j=1}^{N}\alpha_j y_j \,\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j|\mathbf{x}_i,\mathbf{x}_j}\left[K(\mathbf{Z}_i,\mathbf{Z}_j)\right],y_i\right) + \lambda\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i y_i\alpha_j y_j \,\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j|\mathbf{x}_i,\mathbf{x}_j}\left[K(\mathbf{Z}_i,\mathbf{Z}_j)\right].$$

Thus, the VE SVM (asymptotically) minimizes expected loss, while the expected kernel SVM minimizes the loss with respect to the expected similarities.

5.1.2 Expected RBF Kernel for Discrete-time Signals

Model the impulse response of the stochastic channel as the random vector $\mathbf{H}$ with mean $\bar{\mathbf{H}}$ and covariance Σ_h, and model random vector $\mathbf{W}$ as zero mean with covariance $\sigma_w^2 I$. Then, a deterministic vector $\mathbf{x}$ propagated through the stochastic channel results in a random observation

$$\mathbf{Z} = \mathbf{H} * \mathbf{x} + \mathbf{W}.$$

Model $\mathbf{Z} \sim p_{Z|x}(\mathbf{z}|\mathbf{x})$ as Gaussian distributed with mean $\bar{\mathbf{Z}}$ and covariance Σ_z:

$$\bar{\mathbf{Z}} = \bar{\mathbf{H}} * \mathbf{x}, \tag{5.2}$$

$$\Sigma_z = \Sigma_h ** \left(\mathbf{x}\mathbf{x}^T\right) + \sigma_w^2 I. \tag{5.3}$$

To derive the expected RBF training kernel, map $\mathbf{x}_i$ and $\mathbf{x}_j$ to $\mathbf{Z}_i$ and $\mathbf{Z}_j$, which are modeled as independent Gaussians with means and covariances as prescribed in (5.2) and (5.3). Then, evaluate the integral in (5.1) for the RBF kernel in (1.5) using the product-of-Gaussians rule given in (A.9) twice to produce

$$K_{\exp}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, \Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I\right).$$

Similarly, the expected RBF test kernel is also derived using the product-of-Gaussians rule in (A.9):

$$\begin{aligned} K_{\exp}(\mathbf{z}, \mathbf{x}_i) &= \int p(\mathbf{z}_i|\mathbf{x}_i) K(\mathbf{z}, \mathbf{z}_i)\, d\mathbf{z}_i \\ &= \mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right). \end{aligned}$$

5.1.3 Expected RBF Kernel for Randomly Filtered Images

Let $\mathbf{H} = \mathrm{vec}\,(\mathcal{H})$ be a random column-stacked vector of a random 2-d point spread function (filter) $\mathcal{H}$, and let $\mathbf{x}$ be a column-stacked image. Then a randomly filtered image can be represented by the random vector $\mathbf{Z}$ where

$$\mathrm{fold}(\mathbf{Z}) = \mathrm{fold}(\mathbf{H}) * * \mathrm{fold}(\mathbf{x}) + \mathrm{fold}(\mathbf{W}),$$

where fold unstacks an $MN \times 1$ column vector into an $M \times N$ matrix (image), and $**$ represents two-dimensional convolution. Using (A.4), the mean $\bar{\mathbf{Z}}$ and covariance Σ_z of $\mathbf{Z}$ may be expressed as

$$\begin{aligned} \mathrm{fold}(\bar{\mathbf{Z}}) &= \mathrm{fold}(\bar{\mathbf{H}}) * * \mathrm{fold}(\mathbf{x}), \quad \text{and} \\ \mathrm{fold}(\Sigma_z) &= \mathrm{fold}(\Sigma_h) * * * * (\mathrm{fold}(\mathbf{x}) \circ \mathrm{fold}(\mathbf{x})) + \sigma_w^2 I, \end{aligned} \tag{5.4}$$

where the tensor outer product $A \circ B$ of matrices A and B yields a 4th order tensor [41, 4], $\mathrm{fold}(\Sigma_h)$ folds the $MN \times MN$ covariance of column-scanned point spread function into a 4-tensor with dimensions $M \times N \times M \times N$, and $* * * *$ is 4-dimensional discrete convolution. Modeling $p_{Z_i|x_i}$ as a Gaussian with mean and covariance in (5.4) conditioned on $\mathbf{x} = \mathbf{x}_i$, and substituting $p_{Z_i|x_i}$ into (5.1) yields the expected RBF kernel, given in Table 5.2.

5.1.4 Expected RBF Kernel with Subband Energy Features

Model the subband energy feature vector $\mathbf{u}_x$ after propagation through a stochastic channel as a Gaussian random vector $\mathbf{U}_z$, where, from (1.7)

$$\mathbf{U}_z = \mathbf{U}_h \cdot \mathbf{u}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{x}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}.$$

Model $\mathbf{W}^f$ as a proper Gaussian complex random vector with $p(\mathbf{w}^f) = \mathcal{N}\left(0, \sigma_w^2 I\right)$, and random subband energy vector $\mathbf{U}_h$ as having mean $\bar{\mathbf{U}}_h$ and covariance Σ_{u_h} (no additional assumptions are needed about the distribution of $\mathbf{H}^f$).

The expected RBF kernel for subband energy features (with test kernel as a special case) is given by:

$$K_{\exp}(\mathbf{u}_{x_i}, \mathbf{u}_{x_j}) = \mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} I\right),$$
$$K_{\exp}(\mathbf{u}_z, \mathbf{u}_{x_i}) = \mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$$

where

$$\bar{\mathbf{U}}_{z_i} = \mathbf{u}_{x_i} \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1},$$
$$\Sigma_{u_{z_i}} = \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right).$$

The covariance $\Sigma_{u_{z_i}}$ is derived from (B.23) in the appendix by substituting $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$ and $\Sigma_{u_x} = 0$, since we condition on $\mathbf{U}_x = \mathbf{u}_{x_i}$.

5.1.5 Unscaled Expected RBF Kernel

Commonly, the standard RBF kernel is implemented without the Gaussian normalization factor, as $K_{\mathrm{rbf}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2}\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$ so that $K_{\mathrm{rbf}}(\mathbf{x}, \mathbf{x}) = 1$. The inclusion of the Gaussian normalization factor $\left(\frac{\gamma}{2\pi}\right)^{d/2}$ is arbitrary, since it represents a global scaling of the similarity measure. Expected RBF kernels, however, have a bandwidth and scaling that are data-dependent, so that it is necessary to include the scaling factor. In preliminary experiments, the author found that the normalization factor was especially sensitive to choices of γ and estimated statistics of $p_{Z|X}$. In fact, the resulting kernel matrices were often poorly conditioned, and in some cases the SVM solver did not converge.

To mitigate these problems the *unscaled expected RBF kernel* K_{uexp} is defined to be the expected RBF kernel without its Gaussian normalization. Then $K_{\text{uexp}}(\mathbf{x}, \mathbf{x}) = 1$, and the resulting kernel matrices are better conditioned and did not lead to computational problems. In addition, there are significant computational savings since it is no longer required to compute the matrix determinant.

The unscaled expected RBF kernel can be expressed $K_{\text{uexp}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{S(\mathbf{x}_i, \mathbf{x}_j)} K_{\text{exp}}(\mathbf{x}_i, \mathbf{x}_j)$, where $S(\mathbf{x}_i, \mathbf{x}_j)$ is the Gaussian scale factor. We prove that this is still a legitimate kernel. First, note that the Gaussian scale factor $S(\cdot, \cdot)$ is positive definite, since it can be written as the inner product in (5.1) with $p_{Z_i|x_i}(\mathbf{z}_i) \stackrel{\triangle}{=} \mathcal{N}(\mathbf{z}_i; 0, \Sigma_{z_i})$ and $p_{Z_j|x_j}(\mathbf{z}_j) \stackrel{\triangle}{=} \mathcal{N}(\mathbf{z}_j; 0, \Sigma_{z_j})$, resulting in

$$S(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I|^{\frac{1}{2}}},$$

where Σ_{z_i} depends on $\mathbf{x}_i$ and Σ_{z_j} depends on $\mathbf{x}_j$. For a set of any N samples, let S be the $N \times N$ positive definite matrix produced by evaluating the kernel $S(\cdot, \cdot)$ for all pairs of the N samples. Since S is positive definite, the Hadamard inverse $S^{\circ -1} = \left[\frac{1}{S_{ij}}\right]$ is also positive definite [31, p. 397]. The positive definite matrix $S^{\circ -1}$ is the kernel matrix formed by $\frac{1}{S(\mathbf{x}_i, \mathbf{x}_j)}$. Then, we conclude that since the Hadamard product $A \cdot B$ of two positive definite matrices is also positive definite [31], the unscaled expected RBF kernel matrix $K_{\text{uexp}} = S^{\circ -1} \cdot K_{\text{exp}}$ is positive definite.

5.1.6 Modeling Channel Dependency

The definition (5.1) assumes that the random feature vectors $\mathbf{Z}_i$ and $\mathbf{Z}_j$ are independent, which implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ were corrupted by different random draws of the channel and noise. A better model is to treat them as being corrupted by the same draw of the random channel and noise, which produces a joint distribution $p_{Z_i, Z_j|x_i, x_j}(\mathbf{z}_i, \mathbf{z}_j)$, and then the expectation in (5.1) becomes

$$\iint p_{Z_i, Z_j|x_i, x_j}(\mathbf{z}_i, \mathbf{z}_j) K(\mathbf{z}_i, \mathbf{z}_j) \, d\mathbf{z}_i \, d\mathbf{z}_j. \tag{5.5}$$

Unfortunately, it is not clear under what conditions (5.5) will be a legitimate kernel. A closed-form solution to (5.5) for subband energy features is derived in Section B.2. We

compared the resulting classifier with the expected kernel classifier for the experiments detailed in Chapter 6 and found no statistically significant differences between the two in any of the experiments.

5.2 Projected RBF Kernels

We propose another channel-robust kernel that is motivated by a recent interpretation of the RBF kernel. Jebara et al. introduced the *probability product kernel*, which essentially replaces training samples with random variables, $\mathbf{x}_i \mapsto \mathbf{X}' \sim p(\mathbf{x}'|\mathbf{x}_i)$, and defines a positive definite kernel as the inner product of these distributions [36]:

$$K_{\text{prob}}(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\triangle}{=} \int p(\mathbf{x}'|\mathbf{x}_i)p(\mathbf{x}'|\mathbf{x}_j)\, d\mathbf{x}'. \tag{5.6}$$

Jebara et al. noted that the standard RBF kernel with bandwidth parameter γ in (1.5) can be derived as a special case of (5.6) by letting $p(\mathbf{x}'|\mathbf{x}_i) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}_i, \frac{\gamma^{-1}}{2} I\right)$ and $p(\mathbf{x}'|\mathbf{x}_j) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}_j, \frac{\gamma^{-1}}{2} I\right)$, and applying the product of Gaussians identity in (A.9). In order for the RBF kernel to have have same bandwidth parameter γ at test time, the test feature vector $\mathbf{x}$ must also be replaced with a random variable with density $p(\mathbf{x}'|\mathbf{x}) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}, \frac{\gamma^{-1}}{2} I\right)$.

To extend (5.6) to our dataset shift problem, we also consider $\mathbf{x}_i$ to be mapped to a Gaussian random feature vector $\mathbf{X}'$, and then propagate $\mathbf{X}'$ through the stochastic channel, resulting in the random vector $\mathbf{Z}'$. The resulting distribution $p(\mathbf{z}'|\mathbf{x}_i)$ of $\mathbf{Z}'$ is not necessarily Gaussian; however, for mathematical tractability we project $p(\mathbf{z}'|\mathbf{x}_i)$ to the nearest Gaussian using Lemma 1.

Lemma 1. Let random vector $\mathbf{Z} \in \mathbb{R}^d$ be drawn from a distribution that has a probability density function p_Z, finite mean $\bar{\mathbf{Z}} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{S}^d_{++}$. Then, the Gaussian distribution that uniquely minimizes KL-divergence with respect to p_Z is given by $\mathcal{N}(\mathbf{z}; \bar{\mathbf{Z}}, \Sigma)$.

Proof. Let $f(\mathbf{m}, R) = \text{KL}\left(p_Z || \mathcal{N}\left(\mathbf{m}, R\right)\right)$ for $\mathbf{m} \in \mathbb{R}^d$ and $R \in \mathbb{S}^d_{++}$. By definition, $\text{KL}\left(p||q\right) = \text{E}_{\mathbf{Z}}\left[\log p(\mathbf{Z})\right] - \text{E}_{\mathbf{Z}}\left[\log q(\mathbf{Z})\right]$, and thus the mean $\mathbf{m}^*$ and covariance R^* we seek

solve

$$\begin{aligned}
&\arg\min_{R\succ 0,\mathbf{m}} f(\mathbf{m},R) = \arg\min_{R\succ 0,\mathbf{m}} -\mathrm{E}_{\mathbf{Z}}\left[\log\mathcal{N}\left(\mathbf{Z};\mathbf{m},R\right)\right] \\
=&\arg\min_{R\succ 0,\mathbf{m}} \log|R| + \mathrm{E}_{\mathbf{Z}}\left[\left(\mathbf{Z}-\mathbf{m}\right)^T R^{-1}\left(\mathbf{Z}-\mathbf{m}\right)\right] \\
=&\arg\min_{R\succ 0,\mathbf{m}} \log|R| + \mathrm{tr}\ \mathrm{E}_{\mathbf{Z}}\left[\left(\mathbf{Z}-\mathbf{m}\right)\left(\mathbf{Z}-\mathbf{m}\right)^T R^{-1}\right].
\end{aligned}$$

Since $f(\mathbf{m},R)$ is convex in R, the minimizer $\mathbf{m}^*$ is found by solving

$$\nabla_m f(\mathbf{m}^*,R) = -2R^{-1}\,\mathrm{E}_{\mathbf{Z}}\left[(\mathbf{Z}-\mathbf{m}^*)\right] = 0,$$

and therefore $\mathbf{m}^* = \bar{\mathbf{Z}}$ is the unique global minimizer since $\mathbf{m}^*$ does not depend on R.

However, $f(\mathbf{m},R)$ is not convex in $\mathbf{m}$, but for fixed $\mathbf{m} = \bar{\mathbf{Z}}$

$$\begin{aligned}
&\arg\min_{R\succ 0} \log|R| + \mathrm{tr}\ \mathrm{E}_{\mathbf{Z}}\left[\left(\mathbf{Z}-\mathbf{m}\right)\left(\mathbf{Z}-\mathbf{m}\right)^T R^{-1}\right] \\
=&\arg\min_{R\succ 0} -\log|R^{-1}| + \mathrm{tr}\ \Sigma R^{-1} \\
=&\arg\min_{Y\succ 0} -\log|Y| + \mathrm{tr}\ \Sigma Y,
\end{aligned}$$

since the change of variables $Y = R^{-1}$ is a bijection from $\mathbb{S}^d_{++}$ onto $\mathbb{S}^d_{++}$. The function $g(Y) = -\log|Y| + \mathrm{tr}\ \Sigma Y$ is strictly convex [6], so that the unique global minimizer is found by solving

$$\nabla_Y g(Y^*) = -{Y^*}^{-1} + \Sigma = 0,$$

so that $Y^* = \Sigma^{-1}$. We conclude that $\mathbf{m}^* = \bar{\mathbf{Z}}$ and $R^* = \Sigma$ uniquely minimize $f(\mathbf{m},R)$. □

Let $\mathcal{N}(\mathbf{z}|\mathbf{x}_i)$ be the projection of $p(\mathbf{z}|\mathbf{x}_i)$ to the nearest Gaussian distribution using Lemma 1. Then, analogous to (5.6), we define the *projected RBF kernel* as

$$K_{\mathrm{proj}}\left(\cdot,\mathbf{x}_j\right) \triangleq \int \mathcal{N}(\mathbf{z}'|\cdot)\mathcal{N}(\mathbf{z}'|\mathbf{x}_j)\,d\mathbf{z}'. \tag{5.7}$$

When evaluating the kernel between a test sample and training sample $K_{\mathrm{proj}}\left(\mathbf{z},\mathbf{x}_j\right)$, we define the distribution $\mathcal{N}(\mathbf{z}'|\mathbf{z})$ needed for (5.7) to be the projection of a random variable $\mathbf{Z}'$ to the nearest Gaussian, where $\mathbf{Z}'$ results from propagating $\mathbf{X}' \sim \mathcal{N}(\bar{\mathbf{X}}', \frac{\sigma^{-1}}{2} I)$ through the stochastic channel, and $\bar{\mathbf{X}}'$ is chosen such that the mean $\mathrm{E}\left[\mathbf{Z}'\right] = \mathbf{z}$ is the observed test sample (see Sections 5.2.1 and 5.2.3 for examples). The kernel K_{proj} is a legitimate kernel because it is always an inner product of two distributions in $\mathbf{z}'$.

We next present the analytic forms of the projected RBF test and training kernels for the same two cases as the expected RBF kernel: discrete-time signal features, image pixel features, and subband energy features.

5.2.1 Projected RBF Kernel for Discrete-Time Signals

Model the random vector $\mathbf{X}' \sim \mathcal{N}(\mathbf{x}, \frac{\gamma^{-1}}{2} I)$, then $\mathbf{Z}' = \mathbf{H} * \mathbf{X}' + \mathbf{W}$ has mean and covariance given by

$$\begin{aligned} \bar{\mathbf{Z}}' &= \bar{\mathbf{H}} * \mathbf{x}, \quad \text{and} \\ \Sigma_{z'} &= \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}}\bar{\mathbf{H}}^T\right) + \Sigma_h ** \mathbf{x}\mathbf{x}^T + \sigma_w^2 I. \end{aligned} \tag{5.8}$$

Then by Lemma 1, the projection $p(\mathbf{z}'|\mathbf{x}_i)$ to the nearest Gaussian distribution $\mathcal{N}(\mathbf{z}'|\mathbf{x}_i)$ yields

$$\mathcal{N}(\mathbf{z}'; \bar{\mathbf{H}} * \mathbf{x}_i, \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}}\bar{\mathbf{H}}^T\right) + \Sigma_h ** \mathbf{x}_i\mathbf{x}_i^T + \sigma_w^2 I).$$

Substituting into (5.7), and simplifying with the product of Gaussians rule given in (A.9) yields

$$K_{\text{proj}}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}\left(\bar{\mathbf{H}} * \mathbf{x}_i; \bar{\mathbf{H}} * \mathbf{x}_j, \gamma^{-1} I ** \left(\Sigma_h + \bar{H}\bar{H}^T\right) + \Sigma_h ** \left(\mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T\right) + 2\sigma_w^2 I\right).$$

To construct $\mathcal{N}(\mathbf{z}'|\mathbf{z})$, we assume that a test sample is the mean of the distribution, $\mathbf{z} = \mathrm{E}[\mathbf{Z}']$, where the random variable $\mathbf{Z}'$ is the projection through the stochastic channel of a random variable $\mathbf{X}'$ with covariance $\frac{\gamma^{-1}}{2} I$. Therefore, $\mathcal{N}(\mathbf{z}'|\mathbf{z})$ has covariance given by (5.8) with $\mathbf{x}$ substituted with Fourier deconvolution $\mathbf{x}^{-1} = \bar{\mathbf{H}}^{-1} * \mathbf{z}$. Then, the projected RBF test kernel given by (5.7) simplifies to the form in Table 5.1.

5.2.2 Projected RBF Kernel for Image Pixel Features

Let $\mathbf{H}$ be a random column-stacked vector of a random 2-d point spread function, and let $\mathbf{X}$ be a random column-stacked image with mean $\mathbf{x}$ and covariance $\frac{\gamma^{-1}}{2} I$. Model random vector $\mathbf{Z}' \sim \mathcal{N}(\mathbf{z}'|\mathbf{x})$ as

$$\text{fold}(\mathbf{Z}') = \text{fold}(\mathbf{H}) ** \text{fold}(\mathbf{X}) + \text{fold}(\mathbf{W}).$$

The mean $\bar{\mathbf{Z}}'$ and covariance R of $\mathbf{Z}'$ may be expressed as

$$\begin{aligned}\operatorname{fold}(\bar{\mathbf{Z}}') &= \operatorname{fold}(\bar{\mathbf{H}}) * * \operatorname{fold}(\mathbf{x}), \quad \text{and} \\ \operatorname{fold}(R_z) &= \operatorname{fold}\left(\frac{\gamma^{-1}}{2} I\right) * * * * \left(\operatorname{fold}\left(\Sigma_h\right) + \operatorname{fold}\left(\bar{\mathbf{H}}\right) \circ \operatorname{fold}\left(\bar{\mathbf{H}}\right)\right) \\ &\quad + \operatorname{fold}(\Sigma_h) * * * * \left(\operatorname{fold}(\mathbf{x}) \circ \operatorname{fold}(\mathbf{x})\right) + \sigma_w^2 I,\end{aligned}$$

Conditioning on $\mathbf{x} = \mathbf{x}_i$, and substituting into (5.7) yields the projected RBF kernel for column-stacked images $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$K_{\text{proj}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathcal{N}\left(\bar{\mathbf{Z}}_i'; \bar{\mathbf{Z}}_j', R_{z_i} + R_{z_j}\right),$$

also shown in Table 5.2. The covariance term for a test image is taken to be $\tilde{R}_z = R_z\big|_{\mathbf{x}=\hat{\mathbf{x}}}$, where $\hat{\mathbf{x}}$ solves $\operatorname{fold}(\mathbf{z}) = \operatorname{fold}(\bar{\mathbf{H}}) * * \operatorname{fold}(\hat{\mathbf{x}})$.

5.2.3 Projected RBF Kernel for Subband Energy Features

For subband energy features, let $\mathbf{U}_x' \sim \mathcal{N}(\mathbf{u}_{x_i}, \frac{\gamma^{-1}}{2} I)$. Then project the distribution of the random variable $\mathbf{U}_{z_i}' = \mathbf{U}_h \cdot \mathbf{U}_x' + \mathbf{U}_w + 2\operatorname{Re}\left\{\mathbf{X}^{f\prime} \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}$ to the nearest Gaussian $\mathcal{N}(\mathbf{u}_z'|\mathbf{u}_{x_i}) = \mathcal{N}(\mathbf{u}_z'; \bar{\mathbf{U}}_{z_i}', R_{u_{z_i}})$, where

$$\bar{\mathbf{U}}_{z_i}' = \bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i} + \sigma_w^2 \mathbf{1}, \tag{5.9}$$

$$R_{u_{z_i}} = \frac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right), \tag{5.10}$$

which follows from (B.23) in the appendix with $\Sigma_{u_x} = \frac{\gamma^{-1}}{2} I$ and $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$.

Then, solving the integral in (5.7), the projected RBF training kernel takes the form

$$K_{\text{proj}}\left(\mathbf{u}_{x_i}, \mathbf{u}_{x_j}\right) = \mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, R_{u_{z_i}} + R_{u_{z_j}}\right).$$

At test time, given an observation $\mathbf{u}_z$, the distribution $p(\mathbf{u}_z'|\mathbf{u}_z) = \mathcal{N}\left(\mathbf{u}_z, \tilde{R}_{u_z}\right)$, where $\tilde{R}_{u_z}$ is $R_{u_{z_i}}$ in (5.10) with $\hat{\mathbf{u}}_x$ substituted for $\mathbf{u}_{x_i}$; $\hat{\mathbf{u}}_x$ satisfies $\mathbf{u}_z = \hat{\mathbf{u}}_x \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1}$:

$$\hat{\mathbf{u}}_x = \frac{\left[\mathbf{u}_z - \sigma_w^2 \mathbf{1}\right]}{\left[\bar{\mathbf{U}}_h\right]},$$

where $\frac{[\mathbf{a}]}{[\mathbf{b}]}$ denotes Hadamard (element-wise) division of $\mathbf{a}$ and $\mathbf{b}$. Then, solving the integral in (5.7), the projected RBF test kernel for subband energy features is

$$K_{\text{proj}}\left(\mathbf{u}_z, \mathbf{u}_{x_i}\right) = \mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right).$$

5.2.4 Unscaled Projected RBF Kernel

Similar to the discussion of the unscaled expected RBF kernel in 5.1.5, we found that the unscaled version of the projected RBF kernel was computationally more efficient and more robust to parameter choices and estimated statistics. Thus, we used the unscaled projected RBF kernel in our experiments in Chapter 6 and unless specifically stated, when referring to the "projected RBF kernel" we mean specifically the unscaled version. Using the same arguments given in 5.1.5, one can show that the unscaled projected RBF kernel is positive definite.

5.2.5 Projected RBF vs. Expected RBF Kernels

The projected RBF and expected RBF kernels differ in the way that the statistics of the channel samples are incorporated, and in the way that the bandwidth parameter γ is used. Comparing the covariance terms in Tables 5.1, 5.2 and 5.3, we observe that the covariance of the expected RBF kernel has the form

$$\Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I,$$

whereas the covariance of the projected RBF kernel is given by

$$\begin{aligned} R_{z_i} + R_{z_j} &= \Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I * * \left(\Sigma_h + \bar{\mathbf{H}}\bar{\mathbf{H}}^T\right), \text{ or} \\ R_{u_{z_i}} + R_{u_{z_j}} &= \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_j \bar{\mathbf{U}}_h^T\right). \end{aligned}$$

Therefore, the training kernels differ only by the matrix weighted by γ^{-1}; they are identical as $\gamma \to \infty$. Since for the projected RBF SVM, γ^{-1} acts as a weight on the statistics of the random impulse response, we expect that the projected SVM may be more sensitive to errors in estimating the probabilistic transformation $p_{Z|X}$ when γ^{-1} is large.

Another key difference between the two kernels is that the projected RBF kernel treats the test vector $\mathbf{z}$ as a realization of a random vector with nonzero covariance. Conversely, the definition of the expected RBF kernel treats the test point as deterministic.

5.2.6 Adapting SVMs Trained on Clean Data to Corrupted Test Features

The presented expected and projected RBF kernels require that statistics (e.g., sample mean and covariance) of the auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$ are available to train the SVM. For each new environment, the SVM must be re-trained using the statistics of the stochastic channel. While we believe that it is optimal to train the SVM for the particular environment, as a practical question we considered whether we could train an SVM without knowing the environment, and only adapt the SVM for the environment at test time.

When training an SVM, one solves for coefficients $\{\alpha_i\}_{i=1}^N$ which determine the contribution of each training sample as shown in (1.4). Notably, some α_i's are set to zero in the training process, removing certain training samples from influencing the classifier.

Suppose that a kernel function $K(\cdot,\cdot)$ is selected for SVM classification. For cases in which re-training the SVM for each new environment is undesirable, we propose the following approach:

1. Train an SVM on the dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with kernel K to obtain the weights $\{\alpha_i\}_{i=1}^N$ and bias b in Eq. (1.4);

2. For a new propagation environment, collect auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$ and compute relevant statistics;

3. Calculate a bias term for the new environment using the KKT conditions of the SVM [30, p. 374]:

$$b' = \frac{1}{N}\sum_{i=1}^{N}\frac{1-\xi_i}{y_i} - \sum_{j=1}^{N}\alpha_j y_j K_{te}(\mathbf{x}_i, \mathbf{x}_j),$$

   where ξ_i are the SVM slack variables, and $K_{te}(\cdot,\cdot)$ is the channel-robust test kernel function. The new bias b' minimizes the average label prediction error over all support vectors.

4. Classify the test sample as the sign of

$$f(\mathbf{z}) = b' + \sum_{i=1}^{N}\alpha_i y_i K_{te}(\mathbf{z}, \mathbf{x}_i).$$

This approach is theoretically sub-optimal, since the weights $\{\alpha_i\}_{i=1}^N$ that minimized empirical risk using the kernel K are not optimally suited to the test kernel K_{te}. Note that the role of the $\{\alpha_i\}_{i=1}^N$ is to weight how important each training sample is in determining a decision boundary, and these relative importance may not change much for the test conditions. Further, re-calculation of the bias term b' grossly adjusts the decision boundary so that at least the label of the support vectors are, on average, predicted accurately.

5.3 Conclusions

In this chapter, the expected and projected RBF kernels were presented as an alternative to the VE methods. The expected RBF and projected RBF kernels were derived for discrete-time signal, images, and subband energy features. The kernels account for the channel or blur that separate training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ from the test sample $\mathbf{z}$ by using a stochastic channel model, where statistics are inferred from the auxiliary channel features $\{\mathbf{h}_i\}_{i=1}^M$.

The expected RBF and projected RBF kernels include a Gaussian normalization term that, in the case of subband energy features, depends on the samples being evaluated. In preliminary experiments, it was found that the Gaussian scaling term was sensitive to channel estimation and lead to poorly conditioned kernel matrices. Therefore, the unscaled expected RBF and projected RBF kernels were proposed and shown to be positive definite functions.

Lastly, an engineering shortcut was proposed in which an SVM is trained on clean training data using a standard RBF kernel, but at test time, the expected RBF or projected RBF kernel is used in an SVM with an updated bias term. Although suboptimal, the shortcut allows one to train an SVM once, and only update the bias term of the discriminant function for each new propagation environment.

Experiments comparing the (unscaled) expected RBF SVM, projected RBF SVM, and the clean-training engineering shortcut for both classifiers are presented in the next chapter.

Chapter 6

CLASSIFYING SOUNDS IN REVERBERANT ENVIRONMENTS

> As [Henry Eyring and Albert Einstein, colleagues at Princeton] walked together they noted an unusual plant growing along a garden walk. Dr. Eyring asked Dr. Einstein if he knew what the plant was. Einstein did not, and together they consulted a gardener. The gardener indicated the plant was green beans and forever afterwards Eyring said Einstein didn't know beans.
>
> S. K. Franz about Dr. Henry Eyring

This chapter presents three experiments to test the accuracy of the subband energy feature-based classifiers developed in this thesis. First, a controlled experiment is presented, in which simulated narrowband signals are injected into an artificial shallow-water bathymetry using audio-realistic propagation software. The second experiment demonstrates that the algorithms can identify individual Bowhead whales by their vocalizations in a noisy, shallow water environment; data were acquired from whale recordings, but by necessity, they are injected into a shallow-water bathymetry using audio-realistic acoustic propagation software. The third experiment involves real recordings in an outdoor reverberant environment. The aim is to identify trumpeters by the sound of their trumpets; this is a particularly challenging problem, since the trumpeters are playing precisely the same note for precisely the same duration.

6.1 Experimental Details

We compare the expected RBF SVM, the expected RBF SVM (clean) trained on uncorrupted training pairs $\{(\mathbf{u}_{x_i}, y_i\}_{i=1}^N$, projected RBF SVM, projected RBF SVM (clean) trained on $\{(\mathbf{u}_{x_i}, y_i\}_{i=1}^N$ and local joint QDA to VE RBF SVM, VE k-NN and to a channel agnostic RBF SVM. All RBF kernels are unscaled.

Given N subband energy feature vectors $\{\mathbf{u}_{x_i}\}_{i=1}^N$ and M auxiliary samples $\{\mathbf{u}_{h_i}\}_{i=1}^M$, we generate VEs for VE RBF SVM and VE k-NN as follows. For each $\mathbf{u}_{x_i}$, we generate M VEs by taking $\mathbf{u}_{x_i}$ with every element of $\{\mathbf{u}_{h_j}\}_{j=1}^M$ to form

$$\mathbf{u}_{z_{ij}} = \mathbf{u}_{h_j} \cdot \mathbf{u}_{x_i} + \sigma_w^2 \mathbf{1},\ j = 1 \dots P. \tag{6.1}$$

For VE RBF SVM, an SVM is then trained from the $M \times N$ VEs; for VE k-NN, k nearest neighbors are chosen among the VEs. We chose to use the noise power σ_w^2 in (6.1) instead of generating Gaussian noise draws, since to incorporate a noise draw $\mathbf{w}^f$, the VE method would also require the Fourier coefficients $\mathbf{x}_i^f$ and $\mathbf{h}_j^f$ as in (1.7), which are not assumed to be provided for any of the classifiers we compare.

The agnostic RBF SVM is trained on $\{\mathbf{u}_{x_i}\}_{i=1}^N$; auxiliary channel feature vectors $\{\mathbf{u}_{h_i}\}_{i=1}^M$ are ignored.

For the standard machine learning problem, the training and test data are normalized using the sample means and standard deviations of the training samples. However, in this research the training and test features are related by the expression in (1.7), so that normalizing would not properly center and scale the test data. If $\mathbf{m}$ and $\mathbf{s}$ are the mean and scale, respectively, of the training data, and $\tilde{\mathbf{U}}_x = \frac{[\mathbf{U}_x - \mathbf{m}]}{[\mathbf{s}]}$ is the random variable describing the normalized training data, then scaling the test data and taking the expectation over $\mathbf{W}^f$ and $\mathbf{U}_h$ yields

$$\begin{aligned}
\tilde{\mathbf{U}}_z = \frac{\left[\mathbf{U}_z - \bar{\mathbf{U}}_h \cdot \mathbf{m}\right]}{[\mathbf{s}]} &\overset{\text{expectation}}{\rightarrow} \frac{\left[\mathbf{U}_x \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1} - \bar{\mathbf{U}}_h \cdot \mathbf{m}\right]}{[\mathbf{s}]} \\
&= \tilde{\mathbf{U}}_x \cdot \bar{\mathbf{U}}_h + \frac{\left[\sigma_w^2 \mathbf{1}\right]}{[\mathbf{s}]} \\
&\neq \tilde{\mathbf{U}}_x \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1},
\end{aligned}$$

so that data normalization distorts the relationship between the test and training data when the noise power is non-negligible.

Though we cannot normalize the data, we can achieve a similar effect by adjusting the RBF kernel bandwidth parameter. For each dataset, the RBF bandwidth parameter γ is cross-validated over a range of length-scales that are related to the inter-sample distances between points. As a heuristic to choosing reasonable values for γ, we introduce a parameter β that chooses γ^{-1} as a multiple of the minimum inter-neighbor distance according to a logarithmic scale:

$$\gamma^{-1} = \left(\frac{d_{\max}}{d_{\min}}\right)^{\beta} d_{\min},$$

where $d_{\min}$ and $d_{\max}$ are respectively the minimum and maximum inter-sample distances of the training set. Thus, for $\beta = 0$, $\gamma^{-1} = d_{\min}$, for $\beta = 1$, $\gamma^{-1} = d_{\max}$, and so on. For cross-validation, we cross-validate β over the set $\beta \in \{-1.5, -1.25, \ldots, 2.25, 2.5\}$. We allow γ^{-1} to be greater than the maximum inter-neighbor distance (for $\beta > 1$) or less than the minimum inter-neighbor distance (for $\beta < 0$) since γ^{-1} plays the role of a regularization parameter in (5.10), and these larger and smaller values are sometimes chosen. The SVM margin penalty C is cross-validated over the set $\{1, 10^1, 10^2, 10^3, 10^4\}$. The k-NN classifier cross-validates over a single parameter $k \in \{1, 3, 5, 9, 17, 33, N\}$ (goes like $2^n + 1$). Since local joint QDA estimates a class-conditional mean and covariance from k_y training samples *per class*, the role of k_y differs from that of k. For local joint QDA, we cross-validated over the number of class-specific neighbors $k_y \in \{5, 9, 17, 33, |\mathcal{X}_y|\}$ and whether to use the maximum-likelihood estimate of the full covariance or assume a diagonal covariance.

For each of the classifiers, a tie for the parameter pairs (or single parameter for k-NN) that achieved the best cross-validation score was settled by choosing among the best performing parameter pairs randomly with equal probability.

6.2 Classifying Narrowband Acoustic Signals in Simulated Bathymetry

The methodology for creating these data is described in Section 4.3. The dataset consists of subband energies of narrowband signals propagating in a shallow water sonar environment. The training data $\{(\mathbf{u}_{x_i}, y_i)\}_{i=1}^{N}$ for each are linearly separable, but were created so that the

classes class separation can be altered, resulting in three datasets of varying class separation: `easy`, `medium` and `hard`. Realistic sonar channel impulse responses were generated from the bathymetry in Figure 4.2 by using audio-realistic acoustic propagation software provided in the Sonar Simulation Toolset (SST) [25].

The experiment is set up as follows. There are $N = 200$ narrowband training signals from two classes. In addition, $M = 20$ channel impulse responses are provided as part of the training data, from which we compute channel subband energy feature vectors $\{\mathbf{u}_{h_i}\}_{i=1}^{M}$. The 1800 test signals were formed by convolving an i.i.d. source signal with a randomly drawn channel impulse responses, generated i.i.d. with the training channel impulse responses. Each of the channel impulse responses was generated by first randomly picking a source location in a simulated bathymetry, then simulating with the Sonar Simulation Toolset (SST, see [25]) the propagation of an impulse from that random source location to a fixed receiver location. White Gaussian noise is added to the propagated signal so that the SNR ranges from -10 to 10 dB. Results for `easy`, `medium` and `hard` are shown in Fig. 6.2.

The experimental setup differs slightly from Section 4.3 and other publications ([2, 34]). Jamieson et al. [34] compared performance of classifiers for a fixed training time, which necessitated utilizing a smaller number M of auxiliary samples for VE than for expected RBF SVM. In this paper, the experiment is set up to compare performance of algorithms when the same data is available to each, regardless of training time. In addition, in [34], leave-one-out-crossvalidation was performed for each combination of M auxiliary features and N training features. We employ ten-fold cross-validation to determine the values of both γ and C (for SVM) over a classifier-agnostic set of values.

6.2.1 Training Time

The expected and projected RBF SVM classifiers incur a training cost of $O\left(N^3\right)$. However, there is a hidden cost in populating the $N \times N$ kernel matrix, since each entry requires computing the inverse of a $d \times d$ non-diagonal matrix (see Table 5.3), so that populating the matrix incurs a cost of $O(N^d d^3)$. As noted previously, the VE RBF SVM incurs a training cost of $O\left(M^3 N^3\right)$ since the dataset has been increased to a factor of M. Populating

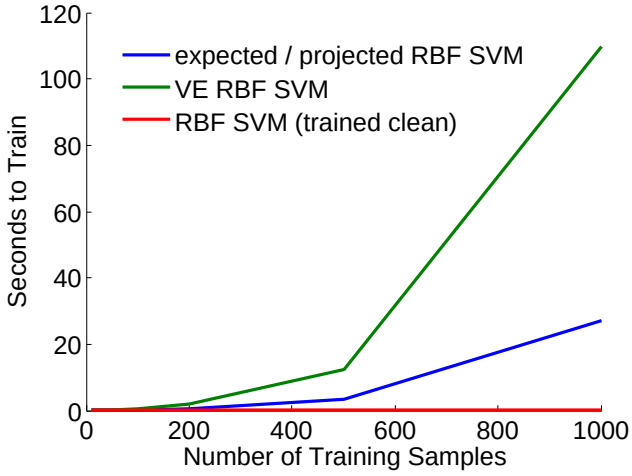


Figure 6.1: SVM training time vs. training set size N for fixed $M = 20$ used in the simulation experiment. Timing results include the time required to populate the kernel matrix.

the $MN \times MN$ RBF kernel is computationally trivial, but if the kernel matrix is precomputed, the expanded dataset places memory restrictions on the usable number M of channel features when the size N of the dataset is large. A plot comparing the training times of expected / projected RBF SVM, VE RBF SVM, and RBF SVM (trained clean) versus the number of training samples is shown in Fig. 6.1—the SVMs were trained using `libsvm` [12] on a 3.2 GHz Intel Core i7 CPU.

6.2.2 Results

Classification results for simulated data in the synthetic bathymetry are shown in Fig. 6.2. For `hard`, clean-train expected SVM is the best overall performer with 95% confidence, followed closely by expected SVM and projected SVM, which are statistically tied. VE SVM gives slightly better performance than local joint QDA. A similar trend holds for `medium`: clean-train expected SVM clean is the best overall performer, followed by projected SVM and VE k-NN (statistically tied overall). For `easy`, VE SVM is the best overall performer, followed closely by expected SVM clean, then VE k-NN and local joint QDA. In all of these

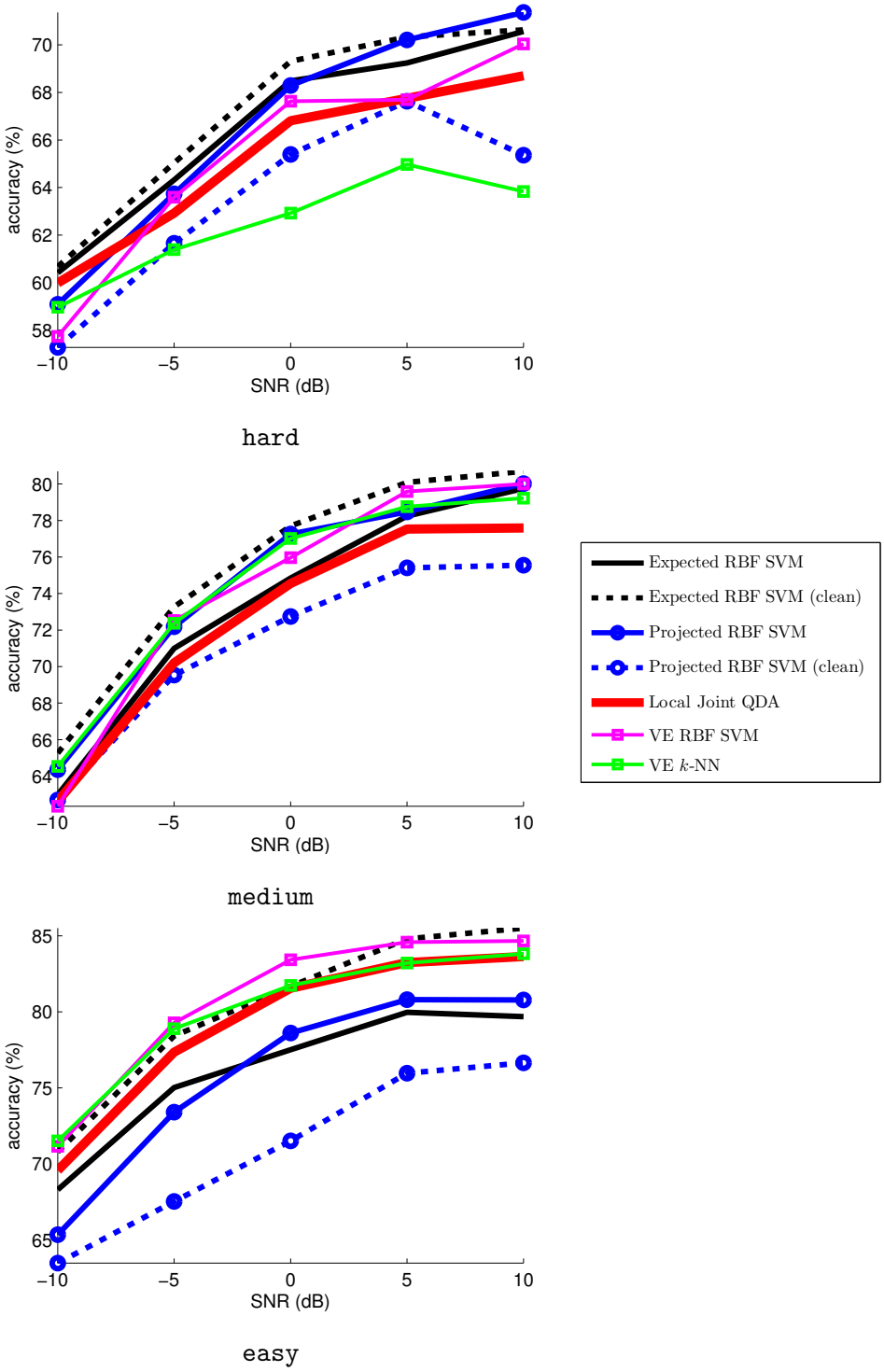


Figure 6.2: Classification accuracy of simulated signals in simulated bathymetry using sub-band energy features. The datasets `hard`, `medium` and `easy` differ in how well the classes are separated in feature space. Note that the accuracy axis for each plot is on a different scale in order to highlight the relative performance of algorithms. RBF SVM (agnostic) achieves an accuracy of 50% for all SNR in each experiment, and is not shown.

experiments, the agnostic SVM, which treats the corrupted test data as though it were not corrupted, produces almost exactly a 50% classification rate, which is as poor as randomly guessing the class label.

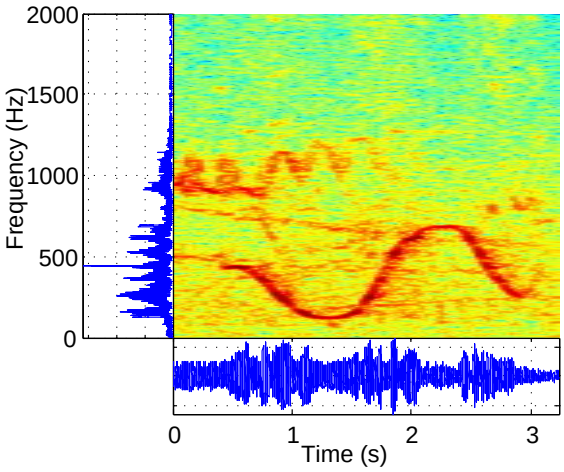
6.3 Classifying Bowhead Whale Songs in Shallow Water

Several end notes of Bowhead whale vocalizations for two individuals were extracted from the MobySound archive [46]. Fifteen vocalizations are available for whale 1, and nine vocalizations are available for whale 2. According to the metadata, the end notes of Bowhead whale songs are relatively stable from year to year. Therefore, we hope to be able to acoustically discriminate between two individuals based on previously recorded vocalizations. Our experimental setup simulates a shallow ocean channel (in comparison to the observation distance) at low SNR. Each of the signals has non-negligible interfering noise from bearded seals, sea ice and banging hydrophone cables [46]. The training signals were recorded in April 1988 near the coast of Point Barrow, Alaska. In our experiments, we inject a holdout set of test signals into randomly drawn locations in the simulated bathymetry shown in Fig. 4.2(a). Example vocalizations for each whale are shown in Fig. 6.3. Four frequencies are selected for subband energy features: the two largest amplitude peaks averaged over signals in class 1 (163 and 258 Hz) and the two largest amplitude peaks for class 2 (588 and 207 Hz). The features do not correspond to strong interfering noise.

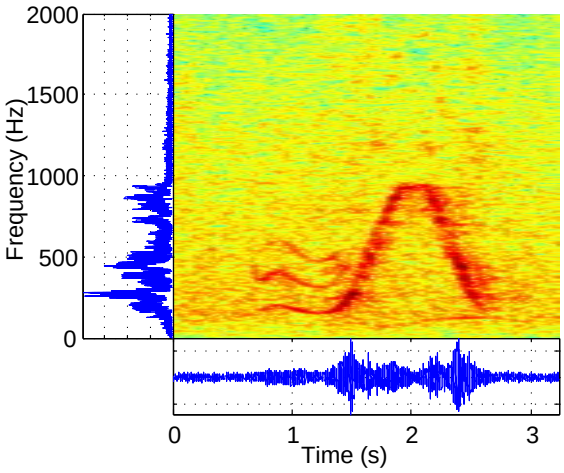
The 24 whale calls are randomly partition into $N = 10$ training signals (5 from each class), and 14 signals from which to generate multipath-corrupted test signals. For the test signals, Gaussian white noise is added so that the SNR of the multipath signal ranges between -10 and 10 dB. Results in Fig. 6.4 were averaged over 1000 i.i.d. training/test partitions.

6.3.1 Results

Classification results for the whale endnotes experiment are shown in Fig. 6.4. Local joint QDA is clearly the best performer over all SNR with statistical significance. It is noteworthy the local joint QDA is less sensitive to the additive noise, than the other classifiers: at -10 dB SNR, it beats the second best performer by 10%. Though close, VE SVM is a slightly better



(a)



(b)

Figure 6.3: Spectrograms of whale song-endnotes for (a) the first Bowhead whale and (b) the second Bowhead whale. The vocalizations of the second whale tend to be more variable, cover a greater dynamic range, and contain stronger harmonic components than the first whale. Notice that the vocalization in (a) contains interfering calls from a bearded seal from about 800 to 1200 Hz.

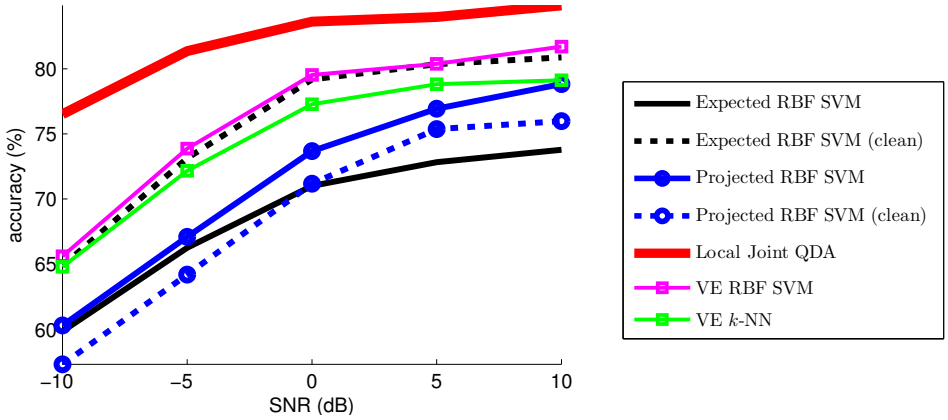


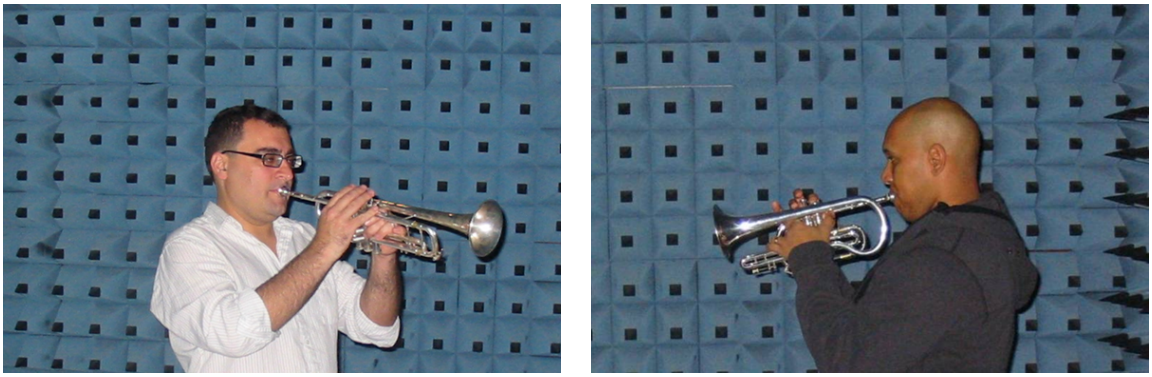
Figure 6.4: Classification accuracy for identifying Bowhead whales in simulated bathymetry by using subband energy features of the end-notes of their songs. RBF SVM (agnostic) achieves an accuracy of $48\% \pm 1\%$ for all SNR, and is not shown.

overall performer for this case than clean-train expected SVM with statistical significance greater than 95%.

We note that the experimental setup for simulated and Bowhead data is different than in [34] and [2]. In [34], the objective was to compare performance for fixed algorithm training time, which necessitated utilizing a smaller number M of auxiliary features for VE than for expected SVM. Here the experiment is set up to compare performance of algorithms when the same data is available to each, regardless of training time. In addition, in [34], leave-one-out-crossvalidation was performed for each combination of M auxiliary features and N training features. In our experiments, we perform ten-fold cross-validation over the N training features, and auxiliary features are drawn randomly from the available M samples as needed. In [2], we note a mislabeling of SNR in the results, so that in comparing results in this paper to results in [2], it erroneously appears that joint QDA outperforms local joint QDA.

6.4 Classifying Trumpeters in Reverberant Environment

The final experiment uses real signals with real multipath corruption. The classifier must discriminate between professional musicians playing the same note on either a trumpet

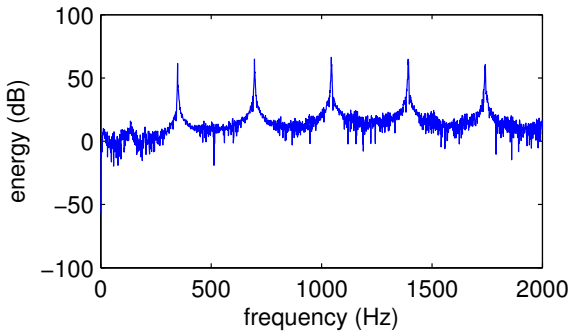


(a) (b)

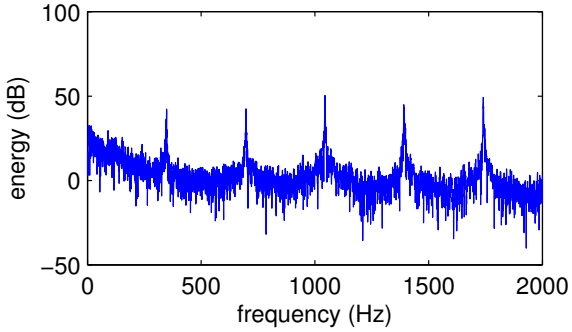
Figure 6.5: (a) Matthew Swihart on trumpet and (b) Edward Castro on cornet in an anechoic chamber.

or cornet in a reverberant environment. The training dataset consists of subband energy features extracted from recordings of two different professional trumpet players. Recordings of Matthew Swihart (`Matt`) and Edward Castro (`Ed`) playing concert F in an anechoic chamber (Fig. 6.5) on their own trumpet and cornet yield four classes: `Matt Trumpet`, `Matt Cornet`, `Ed Trumpet`, `Ed Cornet`. Each of the recordings was clipped to be precisely 1 second in duration. Using the four classes we constructed six different classification problems: `Ed Cornet` vs `Ed Trumpet`, `Matt Cornet` vs `Matt Trumpet`, `Matt Trumpet` vs `Ed Trumpet`, `Matt Cornet` vs `Ed Cornet`, `Matt Trumpet` vs `Ed Cornet`, and `Matt Cornet` vs `Ed Trumpet`.

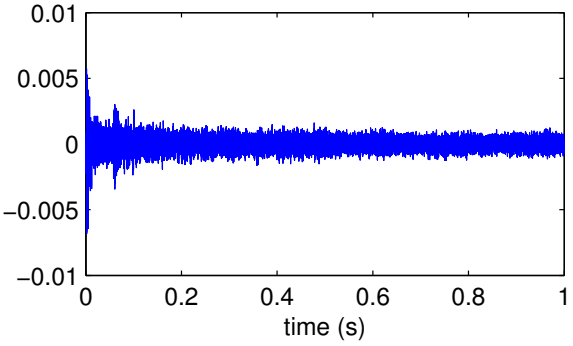
Test signals were recorded in an outdoor semi-enclosed breezeway with a reverberation length of about 1 second. For a controlled experiment, each anechoic signal was played back in the breezeway through high-quality speakers in a fixed location, as well as quadratic chirps to estimate the channel's impulse response. These signals were recorded at four locations with the same recorder, in stereo at a 48kHz sample rate and 16 bits per sample for exactly 2 seconds. When classifying $z[n]$ that corresponds to an anechoic signal $x[n]$, features of $x[n]$ and its stereo pair were excluded from the training set. An example training signal, test signal and an estimated impulse response are shown in Figure 6.6. Figure 6.7 shows a scatterplot of the training and test data on logarithmic axes; the dataset shift is obvious.



(a)



(b)



(c)

Figure 6.6: (a) The energy spectrum of concert F played by Ed on the cornet in the anechoic chamber; (b) the energy spectrum of a test signal generated by playing back the recorded note in an echo chamber; and (c) an impulse response estimated by probing the outdoor breezeway with a quadratic chirp.

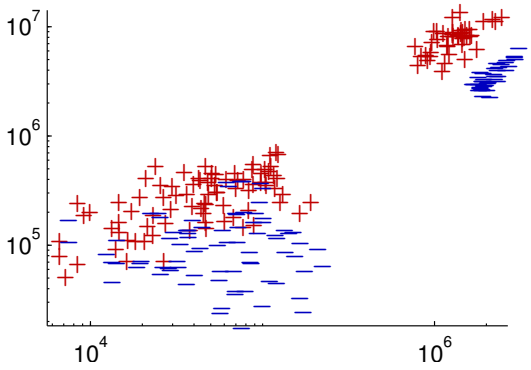


Figure 6.7: Training (upper right) and test features (lower left)—corresponding to subband energies at fundamental and first harmonic—plotted together on a log-log plot, where `Ed Cornet` is denoted by - and `Matt Trumpet` is denoted by +.

Features were taken to be the subband energies at the frequencies

$$\mathbf{f} = \begin{bmatrix} 349 & 698 & 1048 & 1397 & 1746 & 347 & 351 \end{bmatrix}^T \text{ (Hz)},$$

corresponding to the fundamental frequency $f_0 = 349$ Hz, the first four harmonics, and $f_0 - 2$ and $f_0 + 2$ to capture the width of the fundamental. Noise energy σ_w^2 was taken to be the median energy level across all frequency bins. Results in Table 6.1 were averaged over the four different locations.

6.4.1 Results

Table 6.1 shows that in all datasets except `Matt Trumpet` v. `Ed Cornet`, expected SVM is the best performer or statistically tied with the best performer, and in `Matt Trumpet` v. `Ed Cornet`, it is the second best performer. Likewise, local joint QDA is the best (or tied for best) performer in all tests—including `Matt Trumpet` vs. `Ed Cornet`, which apparently was the most challenging comparison—except `Matt Cornet` v. `Matt Trumpet`, for which it is the second best classifier. VE SVM is tied with expected SVM as the best performer in `Matt Cornet` v. `Ed Cornet`. Projected SVM and clean-train projected SVM—which yield similar results in most experiments—perform better than VE SVM in 3 experiments. The clean-train expected SVM classifier does not perform well on the trumpet/cornet experiments.

Table 6.1: Trumpet Playback Results Averages. Bolded items in each column are statistically tied with 95% confidence using a one-sided Wilcoxon sign rank test.

	Matt Cornet v. Ed Cornet	Matt Trumpet v. Ed Trumpet	Matt Cornet v. Matt Trumpet	Ed Cornet v. Ed Trumpet	Matt Cornet v. Ed Trumpet	Matt Trumpet v. Ed Cornet
Expected RBF SVM	**74.8**	**72.0**	**82.9**	**60.3**	**82.3**	60.4
Expected RBF SVM (clean)	68.6	52.0	57.9	52.4	57.3	60.4
Projected RBF SVM	57.6	56.4	73.7	57.4	68.0	59.1
Projected RBF SVM (clean)	55.2	63.9	76.9	57.1	69.8	58.1
Joint QDA	**73.2**	**72.0**	80.7	**61.2**	**81.1**	**65.4**
VE RBF SVM	**73.0**	68.0	72.5	51.0	66.9	61.1
VE k-NN	71.0	56.4	77.7	**60.4**	73.8	60.4
RBF SVM (agnostic)	51.2	49.7	57.7	51.3	52.5	**62.8**

6.5 Summary of Experimental Results

First, we note that the agnostic RBF SVM, which ignores the channel, fails miserably for almost all of these experiments, and thus some form of channel-adaptation should be used. However, given how poor the channel estimates were for these experiments (especially for the trumpet classification), the classification gains produced by the adapted methods were pleasantly surprising.

The clean-train expected/projected SVM classifiers have the least channel adaptation. They use the SVM coefficients $\{\alpha_i\}$ trained on the clean training data, and only adapt the kernel at test time to attempt to better model similarity between the test sample and training sample. Both clean-train SVMs do significantly better than the agnostic over the datasets, suggesting that adapting only the kernel is worthwhile. The clean-train projected SVM generally performs worse than the clean-train expected SVM. We hypothesized that the expected SVM would always do better than its clean-train counterpart because its coefficients were trained for the test-environment. Surprisingly, for both experiments using the bathymetry to generate sonar impulse responses, the expected RBF SVM *clean* consistently does better than the expected RBF SVM. However, the expected RBF SVM clean does not do well at the trumpet identification. We suspect that this is because the channels impulse response estimates were not of sufficient quality, to which the clean-train algorithms are sensitive.

Overall, the expected and projected kernels performed similarly, with very comparable performance on the simulation results, a win for projected on the real whale data, and a win for expected on the real trumpet data. Based on comparing the mathematical formulas of the expected and projected RBF, we hypothesize that the projected RBF kernel will be more sensitive to the quality of the channel estimation errors. This hypothesis is supported by the experimental results showing that projected RBF SVM performs comparatively worse than other algorithms with the poor-quality estimates of the reverberation channels in the breezeway, whereas it was competitive in experiments with simulated multipath.

The VE methods performed poorly on the trumpet data relative to the expected SVM, but comparably when given the simulated bathymetry channels. This may be because

the regularization inherent in expected SVM by aggregating the example channels into a channel mean and covariance is more helpful when the channel examples are poor, as in the case of the trumpet data. Further, the VE methods do better relative to the proposed expected/projected kernels on problems where the classes are easier to separate: such as the `easy` simulation and the whale problem. But the VE methods do worse relative to the proposed expected/projected kernels on problems where the classes are harder to separate: such as the `hard` simulation and the trumpet problem. The clean-train expected SVM performs comparably to the VE SVM for all the simulated channel problems despite taking orders of magnitude less training time, but performs slightly worse on the trumpet datasets.

We found local joint QDA to be the most robust classifier. Its performance on the trumpet data is tied for best, it is the clear winner in distinguishing the whales, and it performs fairly well with the narrowband signal experiment. Also, compared to the SVM classifiers, we found the local joint QDA method to be much more robust to the choice of its cross-validation parameters.

6.6 Conclusion

This chapter has compared several classifiers to address the dataset shift problem that occurs in signal classification problems when the differences between training and test conditions can be modeled by a linear time-invariant channel and additive Gaussian white noise.

Experiments with simulated and real data revealed the following trends. On easier problems, where the classes were well-separated and the channels were realistic but simulated, the VE methods performed well. On harder problems, where the classes are less well-separated and the channel estimation was poorer, the expected and projected kernel SVMs performed better. In particular, the expected kernel seemed most robust to non-idealistic conditions, but less able to take advantage of good conditions. In addition, not only are the expected/projected kernels theoretically much faster to train due to the $O(N^3)$ complexity of the SVM training procedure, that in practice with even relatively small sample sizes they were significantly faster to train.

Under cleaner conditions, it was surprising to see that the clean-train expected SVM outperformed the expected SVM, and was often the best performer of all the considered

methods. Notably, clean-train SVMs are the fastest SVMs to train. Throughout, the local joint QDA method performed consistently well, was robust to parameter choices and estimates, and is trivial to train. Modifying local joint QDA for the problem of estimating Gaussian parameters in high dimensions with few training samples is straightforward by applying results the in [60].

While further experimental studies with a wider variety of channels and data are needed, our advice to practitioners based on the experimental evidence we have is to use the clean-train expected SVM is not considered too severe, and—since they seem to be more forgiving classifiers—to use the expected SVM or local joint QDA if the channels are thought to be highly corrupting or poorly estimated.

Chapter 7

EXTENSIONS AND CONCLUSIONS

> A scientist's accomplishments are equal to the integral of his ability integrated over the hours of his effort.
>
> Dr. Henry Eyring, chemist

A summary of the contributions of this thesis are presented in Section 7.1. Section 7.2 includes a discussion of the limitations of the framework and algorithms presented in previous chapters. Finally, the thesis concludes in Section 7.3 with suggested directions for future work.

7.1 Contributions

This thesis focused on deriving classifiers for test samples that are corrupted by a noisy linear time-invariant system given clean training examples. This traditionally signal-processing problem was recast in a machine learning framework in Chapter 1, and by doing so, a previously unpublished flavor of dataset shift was formally proposed. The classifiers introduced to solve this dataset shift problem are broadly categorized by their design to perform *joint deconvolution and classification*, and differentiated by the modeling assumptions and requirements.

Signal-based classifiers were presented in Chapter 3. The joint MAP classifier jointly estimates the signal, channel estimate and class label via the MAP rule. Although the objective function is not convex, reasonable deconvolution estimates and class labels are produced in practice. If the goal is only to produce an estimate of the class label, then better performance is achieved by probabilistically accounting for the convolution model using the joint QDA classifier. The joint QDA classifier models the class-conditional test signal as Gaussian so that an explicit model for the impulse response is not needed.

The joint MAP and joint QDA signal-based classifiers may be difficult to apply in practice, since both algorithm require the inversion of an $L \times L$ covariance matrix (for discrete-time signals of length L). Nevertheless, both algorithms outperformed the Cabrelli blind deconvolution method followed by matched-filter classification, and motivate the value in considering deconvolution jointly with classification.

The joint QDA classifier was derived in Chapter 4 for subband energy features, which are useful discrimining features for the passive acoustic experiments in Chapter 6. The Gaussian assumption underlying joint QDA was relaxed to reduce model bias; instead the assumption is that $p(\mathbf{z}|y)$ is *locally* Gaussian. Joint QDA is a good utility classifier: it exhibited good performance across the datasets considered in this thesis, was robust to specific assignment of the class neighborhood parameter k_y, and exhibited robustness to noisy channel estimates. As a generative classifier, it can be naturally extended to report confidence, incorporate prior information, and operate as a one-class classifier. Since the class-conditional mean and covariance must be estimated from samples, methods for robustly estimating Gaussian distributions from few samples, as in [60], make it generally applicable.

The joint QDA derivation for second order statistics of subband energy features in the appendix (Section B.1) was also a basis for deriving channel-robust kernels in Chapter 5. The expected kernel measures the average similarity between training samples, averaged over the stochastic channel model $p(\mathbf{z}|\mathbf{x})$. The projected RBF kernel was derived by "projecting" a Gaussian RBF through the stochastic channel model. It was found that the Gaussian scaling factors of the kernel functions were sensitive to channel estimation errors. So *unscaled* versions of each kernel were presented, and were shown to be positive definite functions. The expected RBF and projected RBF kernels were proposed as an alternative to the RBF SVM trained on VEs, since training time is dramatically reduced for a large number M of auxiliary channel features. Experiments showed that not only did the proposed kernels require less time to train, but more often than not exhibited superior performance. The robust kernels presented in Chapter 5 represent a richer development of an earlier version of the expected kernel presented in [34].

For additional savings in training time, an engineering shortcut was proposed for training an SVM once on clean data, then adjusting the bias term and kernel in the discriminant

function for each new environment encountered; though suboptimal, no retraining is required. Experiments showed that the clean-trained classifiers performed well on simulated data, but did not perform as well on real data.

In all, five subband feature-based classifiers were presented and tested in experiments: expected RBF SVM, expected RBF SVM clean, projected RBF SVM, projected RBF SVM clean, and the sole generative classifier, local joint QDA. Across all datasets, the local joint QDA classifier gave the best performance, and is the simplest to train. The expected RBF SVM also gave very good performance, and seemed to be especially forgiving of low-quality channel impulse response estimates in the real-data experiment. The projected RBF SVM was shown to be susceptible to choices of γ and channel statistics.

7.2 Limitations

The problem setup assumes two criteria that may not be met in practice. First, clean training samples $\{\mathbf{x}_i\}_{i=1}^N$ are assumed. In practice, it may be more likely that training samples of the form $\{\tilde{\mathbf{z}}_i\}_{i=1}^N$ are available, where $\tilde{\mathbf{z}}_i = \tilde{\mathbf{h}}_i * \mathbf{x}_i + \mathbf{w}_i$; that is, the training samples themselves are corrupted with a channel and noise that may be unique to that particular training sample. The challenge then is to collect auxiliary compound-channel features $\{\mathbf{g}_i\}_{i=1}^M$ distributed i.i.d. from the same distribution as $\mathbf{g} = \tilde{\mathbf{h}}^{-1} * \mathbf{h}$, so that $\mathbf{z} = \mathbf{g} * \tilde{\mathbf{z}} + \tilde{\mathbf{w}}$ accounts for the effects of the training and test corrupting channels. Secondly, it is assumed that auxiliary channel features $\{\mathbf{h}_i\}_{i=1}^M$ are drawn i.i.d. from the same distribution as the true corrupting impulse response $\mathbf{h}$, but often, the auxiliary channel features and the test channel may suffer from their own form of dataset shift. Robustness to these conditions has not been rigorously explored, but in real data experiments where both of these conditions have been violated to some degree, reasonable performance was achieved by the classifiers.

Each classifier presented in this thesis has been derived for two kinds of features: discrete time signal features and subband energy features. Extending the classifiers to feature maps that are a linear function of the signal, $\phi(\mathbf{x}) = A\mathbf{x}$, is straightforward. However, the features that a particular researcher finds compelling for class discrimination are many. For example, this thesis has not derived classifiers that use cepstral coefficients as features,

but their use as discriminating features for, e.g., speech recognition is well documented [8]. (In the noiseless case, the relationship would be trivial, since for cepstral coefficient vector $\mathbf{z}^c = \mathbf{h}^c + \mathbf{x}^c$ is a simple additive model. But noise complicates the relationship.) Alas, for every kind of feature that one may wish to employ, a new derivation of the classifier or kernel is required. This may be intractable for choices of features that are not simple functions of the underlying signal.

On a related note, this thesis has not explored using several feature types. For traditional machine learning methods, the "features" may consist of a mixed bag of descriptions about the object of interest. But, in this thesis, only a homogeneous feature choice for the elements of $\mathbf{x}$ are considered. Since each of the classifiers in this thesis requires second-order statistics of the test features, a first step towards a "mixed bag" is to derive the cross-correlation between, for example, discrete-time features and subband energy features.

Although the expected kernel in Chapter 5 was defined for any kernel K, derivations are presented only for the RBF kernel, which is the most popularly used kernel. Derivations for the linear kernel were presented with an early version of the expected kernel in [34].

7.3 Future Work

Local Joint QDA, the expected kernel SVM and the projected RBF kernel SVM proposed in this thesis solve the dataset shift problem by implicitly transforming the training distribution p_{XY} to p_{ZY} using a forward convolution model: they are *z-space classifiers*. Certainly, one might derive other *z-space* classifiers the leverage the tools developed in this thesis.

One of the chief limitations listed above is that the conditional distribution $p_{Z|X}$ must be modeled from the function relationship of $\mathbf{X}$ and $\mathbf{Z}$ for each new choice of features. An ambitious remedy is to estimate $p_{Z|X}$ directly from the data. This might be accomplished in a semi-supervised learning paradigm, in which a batch of unlabeled test points $\{\mathbf{z}_i\}_{i=1}^{Q}$ is provided in conjunction with the labeled training points $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ [68]. For covariate shift, kernel mean matching is used to estimate the ratio $\frac{p_{te}(\mathbf{x},y)}{p_{tr}(\mathbf{x},y)}$ from the data, without explicitly modeling the distributions [53, Ch. 8].

A key motivation for using a forward convolution model is to avoid ill-posed deconvolution. As noted in Section 1.2.1, the convolution operator has a null space that corresponds

to zeros of the system's frequency response; multipath channels contain zeros due to destructive interference. A similar motivation comes from the data processing theorem [13, Theorem 2.8.1] which states (in the notation of this thesis) that if random variable $\hat{\mathbf{X}}$ is estimated from $\mathbf{Z}$, where $\mathbf{Z}$ has been generated from an underlying source $\mathbf{X}$, as a Markov chain $\mathbf{X} \rightarrow \mathbf{Z} \rightarrow \hat{\mathbf{X}}$, then

$$I(\mathbf{X}; \mathbf{Z}) \geq I(\mathbf{X}; \hat{\mathbf{X}}),$$

where $I(\mathbf{X}; \mathbf{Z})$ is the mutual information between $\mathbf{X}$ and $\mathbf{Z}$. In other words, processing the data (for example, finding a deconvolution estimate $\hat{\mathbf{x}}$) can only destroy information.

Nevertheless, there are practical reasons why one may wish to classify an estimate $\hat{\mathbf{x}}$ rather than the observation $\mathbf{z}$. For example, $\mathbf{z}$ may be of very high dimension when compared to $\hat{\mathbf{x}}$. Or $\mathbf{x}$ may represent the output of another system, such as a Kalman tracker [14]. The following section presents an overview of *robust x-space classifiers*—developed in parallel with the *z-space classifiers* in this thesis—as an alternative approach to solving the dataset shift problem, elements of which were published in [3].

7.3.1 Robust Classifiers for Probabilistic Deconvolution

In this section, let us posit the existence of $p(\mathbf{x}|\mathbf{z})$ that represents the density over all possible deconvolution estimates of $\mathbf{z}$. A draw $\hat{\mathbf{x}}$ from the distribution $p(\mathbf{x}|\mathbf{z})$ would represent a feasible deconvolution estimate of $\mathbf{z}$. It is assumed that a density $p(\mathbf{x}|\mathbf{z})$ can be inferred from the test sample $\mathbf{z}$, training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, and auxiliary training pairs $\{\mathbf{h}_i\}_{i=1}^{M}$. Then, the objective of robust x-space classifiers is to assign a class label to $p(\mathbf{x}|\mathbf{z})$ given training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. A depiction of this problem is shown in Figure 7.1.

While theoretically convenient, it is quite challenging to model $p(\mathbf{x}|\mathbf{z})$ in practice. In this section, a very simple case is considered, in which it is assumed that the test sample and unknown (true) sample are related by

$$\mathbf{z} = H\mathbf{x} + \mathbf{w},$$

where the matrix $H = \text{convmtx}(\mathbf{h})$ is a known. This is a degenerate case of the dataset shift problem depicted in Figure 1.2, in which the set of auxiliary channel features has a

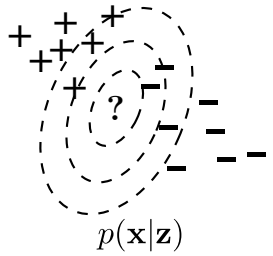


Figure 7.1: Depiction of the problem setup for robust x-space classifiers, in which the labeled training examples (+ and −) are used to infer the class label of a distribution over deconvolution estimates, $p(\mathbf{x}|\mathbf{z})$, represented by its mean (marked **?**) and standard deviation contours.

single element $\mathbf{h}$. For additional simplicity, it is assumed that $\mathbf{x}$ and $\mathbf{z}$ are draws from a jointly Gaussian distribution,

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}; \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\Sigma & \Sigma H^T\\ H\Sigma^T & H\Sigma H^T + \sigma_w^2 I\end{bmatrix}\right),$$

so that

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(\mathbf{x}; \mathbf{m}_{x|z}, \Sigma_{x|z}) \\ &= \mathcal{N}\left(\mathbf{x}; \Sigma H^T \left(H\Sigma H^T + \sigma_w^2 I\right)^{-1} \mathbf{z}, \left(\Sigma^{-1} + \frac{H^T H}{\sigma_w^2}\right)^{-1}\right), \end{aligned} \tag{7.1}$$

which follows from conditional independence of jointly Gaussian random variables [61]. Note that the mean of the distribution $p(\mathbf{x}|\mathbf{z})$ is $\mathbf{m}_{x|z} = \Sigma H^T \left(H\Sigma H^T + \sigma_w^2 I\right)^{-1} \mathbf{z}$, which is precisely the linear minimum mean-squared error (LMMSE or Wiener) deconvolution of $\mathbf{z}$ [47].

With this theoretically convenient setup, in the following subsections, closed form solutions are derived for local QDA and SVM classifiers that operate on the deconvolution density $p(\mathbf{x}|\mathbf{z})$.

Robust Local QDA

First, consider the following definitions.

Definition 2. *Expected Maximum Likelihood (Expected ML) Rule.* Given the distribution $p(\mathbf{x}|\mathbf{z})$, define the expected ML rule as

$$\begin{aligned} y^* &= \arg\max_y \mathrm{E}\left[\mathbf{X}|\mathbf{z}\right] p(\mathbf{X}|y) \\ &= \arg\max_y \int p(\mathbf{x}|y)p(\mathbf{x}|\mathbf{z})\, d\mathbf{x}. \end{aligned} \tag{7.2}$$

Equation (7.2) generalizes the traditional ML rule in the case of no uncertainty for which $p(\mathbf{x}|\mathbf{z}) = \delta(\mathbf{x} - \mathbf{z})$.

Definition 3. *Expected Nearest Neighbor to a Random Test Sample.* Given a random variable $\mathbf{X}$ with finite mean $\mathbf{m}$ and covariance Σ, and given training samples $\{\mathbf{x}_i\}_{1=1}^N$, define the nearest neighbor of $\mathbf{X}$ to be $\mathbf{x}_\ell$ where ℓ solves

$$\begin{aligned} \ell &\stackrel{\triangle}{=} \arg\min_{i=1,\ldots,N} \mathrm{E}_{\mathbf{X}}\left[\|\mathbf{X} - \mathbf{x}_i\|^2\right] \\ &= \arg\min_{i=1,\ldots,N} \mathrm{E}_{\mathbf{X}}\left[(\mathbf{X} - \mathbf{x}_i)^T (\mathbf{X} - \mathbf{x}_i)\right] \\ &= \arg\min_{i=1,\ldots,N} \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{m} + \mathrm{tr}\,\Sigma + \mathbf{m}^T\mathbf{m} \\ &= \arg\min_{i=1,\ldots,N} \|\mathbf{x}_i - \mathbf{m}\|^2, \end{aligned}$$

since Σ does not depend on i.

Note that Definition 3 differs from Definition 1 in Section 4.2, since in Definition 3, only the test point is a random variable; in Definition 1, only the training points are random variables.

Local robust QDA for classifying a deconvolution density is derived using Definition 2, where $p(\mathbf{x}|y)$ is assumed to be locally Gaussian with (local) class-conditional mean $\mathbf{m}_{x|y}$ and covariance $\Sigma_{x|y}$ estimated from the k_y expected nearest neighbors of class y using Definition 3. Then, since $p(\mathbf{x}|\mathbf{z})$ is Gaussian,

$$\begin{aligned} y^* &= \arg\max_y \int p(\mathbf{x}|y)p(\mathbf{x}|\mathbf{z})\, d\mathbf{x} \\ &= \arg\max_y \int \mathcal{N}(\mathbf{x}; \mathbf{m}_{x|y}, \Sigma_{x|y})\mathcal{N}(\mathbf{x}; \mathbf{m}_{x|z}, \Sigma_{x|z})\, d\mathbf{x} \\ &= \arg\max_y \mathcal{N}(\mathbf{m}_{x|z}; \mathbf{m}_{x|y}, \Sigma_{x|y} + \Sigma_{x|z}) \end{aligned}$$

using the product of Gaussians rule in (A.9), and where $\mathbf{m}_{x|z}$ and $\Sigma_{x|z}$ are given in (7.1). Thus, the robust local QDA classifier is precisely the same form as a local QDA classifier operating on a deconvolution estimate $\mathbf{m}_{x|z}$, but with class-conditional covariance regularized by $\Sigma_{x|z}$.

A Bayesian approach to estimate class-conditional Gaussians from few samples was presented in [60], which applied to the robust local QDA classifier results in the *robust local BDA* classifier presented in [3].

Expected Discriminant for Robust SVM

Consider a discriminant classifier, such as an SVM, that classifies a test sample $\mathbf{x}$ as $\text{sgn}(f(\mathbf{x}))$, and define the expected discriminant as follows.

Definition 4. *Expected discriminant.* Let random vector $\mathbf{X}$ be distributed as the deconvolution density $p(\mathbf{x}|\mathbf{z})$. Then, define the expected discriminant as

$$\mathrm{E}_{\mathbf{X}|\mathbf{z}}\left[f(\mathbf{X})\right].$$

For the SVM with RBF kernel, $f(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$ so that for $p(\mathbf{x}|\mathbf{z})$ given in (7.1), the expected discriminant becomes

$$\begin{aligned}
\mathrm{E}_{\mathbf{X}|\mathbf{z}}\left[f(\mathbf{X})\right] &= b + \int p(\mathbf{x}|\mathbf{z}) \sum_{i=1}^{N} \alpha_i y_i K\left(\mathbf{x}, \mathbf{x}_i\right)\, d\mathbf{x} \\
&= b + \sum_{i=1}^{N} \alpha_i y_i \int p\left(\mathbf{x}|\mathbf{z}\right) K\left(\mathbf{x}, \mathbf{x}_i\right)\, d\mathbf{x} \\
&= b + \sum_{i=1}^{N} \alpha_i y_i \int \mathcal{N}\left(\mathbf{x}; \mathbf{m}_{x|z}, \Sigma_{x|z}\right) \mathcal{N}\left(\mathbf{x}; \mathbf{x}_i, \gamma^{-1} I\right)\, d\mathbf{x} \\
&= b + \sum_{i=1}^{N} \alpha_i y_i \, \mathcal{N}\left(\mathbf{m}_{x|z}; \mathbf{x}_i, \Sigma_{x|z} + \gamma^{-1} I\right),
\end{aligned}$$

using the product of Gaussians rule in (A.9). Thus, if the SVM is trained using an RBF kernel with bandwidth (inverse covariance) parameter γ, it may be applied to the deconvolution density $p(\mathbf{x}|\mathbf{z})$ by using a new kernel with covariance $\Sigma_{x|z} + \gamma^{-1} I$.

7.3.2 Forward Model or Inverse Model?

To conclude, there are merits and disadvantages of both *x-space* and *z-space* classifiers. In both cases, a key challenge is modeling the relationship of p_{XY} to p_{ZY}, although modeling is arguably more difficult for *x-space* classifiers. For *z-space* classifiers, p_{ZY} is factored as $p_{Z|X}\, p_{XY}$ so that $p_{Z|X}$ is the distribution to estimate. For example, the class-conditional distribution in joint QDA can be expressed as $p(\mathbf{z}|y) = \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x}|y)\ d\mathbf{x}$; likewise, the expected kernel and projected RBF kernel employ $p_{Z|X}$. In this thesis, the functional relationship between $\mathbf{X}$ and $\mathbf{Z}$ was used to estimate first- and second-order moments of $p_{Z|X}$.

For *x-space* classifiers, the distribution of interest is $p_{X|Z}$, which due to ill-posed deconvolution is generally difficult to model. As an example, for subband-energy features, estimating $p_{X|Z}$ as a Gaussian requires estimating the conditional mean and variance of [rewriting (1.7) in terms of $\mathbf{u}_x$]:

$$\mathbf{u}_x = \frac{\left[\mathbf{u}_z - \mathbf{u}_w - 2\,\mathrm{Re}\left\{\mathbf{x}^f \cdot \mathbf{h}^f \cdot \mathbf{w}^{f*}\right\}\right]}{[\mathbf{u}_h]},$$

which exhibits circular dependence, since $\mathbf{u}_x = \mathbf{x}^f \cdot \mathbf{x}^{f*}$.

However, if $p_{X|Z}$ can be modeled, then *x-space* classifiers are arguably more convenient than *z-space* classifiers. For example, the *z-space* RBF SVMs in Chapter 5 require that the SVM be retrained for each new environment, unless an engineering shortcut is employed. In contrast, an *x-space* SVM need only be trained once, as in Section 7.3.1; different manifestations of the environment are captured by $p(\mathbf{x}|\mathbf{z})$, which is needed only at test time.

As is so often the case in signal processing and statistical learning, the central issue in either approach is modeling.

BIBLIOGRAPHY

[1] Y. S. Abu-Mostafa. Learning from hints in neural networks. *J. Complexity*, 6(2):192–198, 1990.

[2] H. S. Anderson and M. R. Gupta. Joint deconvolution and classification with applications to passive acoustic underwater multipath. *J. Acous. Soc. Am.*, 124(5):2973–2983, November 2008.

[3] H. S. Anderson and M. R. Gupta. Classifying linear system outputs by robust local Bayesian quadratic discriminant analysis on linear estimators. *Proc. IEEE SSP*, 2009.

[4] B. W. Bader and T. G. Kolda. MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32:635–653, 2006.

[5] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *J. Mach. Learn. Res.*, 10:2137–2155, 2009.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] M. K. Broadhead and L. A. Pflug. Performance of some sparseness criterion blind deconvolution methods in the presence of noise. *J. Acoust. Soc. Am.*, 107(2):885–893, Februrary 2000.

[8] L. Buera, E. Lleida, A. Miguel, A. Ortega, and O. Saz. Cepstral vector normalization based on stereo data for robust speech recognition. *IEEE Trans. Audio, Speech and Lang. Proc.*, 15(3):109–1113, March 2007.

[9] C. A. Cabrelli. Minimum entropy deconvolution and simplicity: A noniterative algorithm. *Geophysics*, 50:394–413, 1984.

[10] E. Candes. Compressive sampling. *Proc. Int. Congress of Math.*, 3:1433–1452, 2006.

[11] D. A. Caughey and R. L. Kirlin. Blind deconvolution of echosounder envelopes. *Proc. IEEE ICASSP*, pages 3149–3152, 1996.

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

[13] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley & Sons, Inc., Hoboken, NJ, 2006.

[14] J. L. Crassidis and J. L. Jundkins. *Optimal Estimation of Dynamic Systems.* CRC Press, 2002.

[15] N. Dasgupta and L. Carin. Time-reversal imaging for classification of submerged elastic targets via Gibbs sampling and the relevance vector machine. *J. Acoust. Soc. Am.*, 117(4):1999–2011, 2005.

[16] D. Decoste and B. Schölkopf. Training invariant support machines. *Mach. Learn.*, 46:161–190, 2002.

[17] R. Duda, P. Hart, and D. Stork. *Patter Classification.* John Wiley & Sons, Inc., 2nd edition, 2001.

[18] J. E. Ehrenberg, T. E. Ewart, and R. D. Morris. Signal-processing techniques for resolving individual pulses in a multipath signal. *J. Acoust. Soc. Am.*, 63(6):1861–1865, June 1978.

[19] T. E. Ewart, J. E. Ehrenberg, and S. A. Reynolds. Observations of the phase and amplitude of individual Fermat paths in a multipath environment. *J. Acoust. Soc. Am.*, 63(6):1801–1808, June 1978.

[20] R. E. Fan, P. H. Chen, and C. J. Lin. Working set selection using the second order information for training SVM. *J. Mach. Learn. Res.*, 6:1889–1918, 2005.

[21] R. L. Field. Transient signal distortion in a multipath environment. *Proc. OCEANS*, pages 111–114, 1990.

[22] J. H. Friedman. Regularized discriminant analysis. *J. Am. Stat. As.*, 84(405):165–175, 1989.

[23] E. K. Garcia, S. Feldman, M. R. Gupta, and S. Srivastava. Completely lazy learning. *IEEE Trans. Knowledge Data Engineering*, 2009.

[24] M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.*, 2:299–312, March 2002.

[25] R. P. Goddard. The Sonar Simulation Toolset, Release 4.6: Science, Mathematics, and Algorithms. Technical Report A352884, University of Washington Applied Physics Lab, 2008.

[26] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Trans. Imag. Proc.*, 12(5), May 2003.

[27] M. R. Gupta and H. S. Anderson. Maximum likelihood signal classification using second-order blind deconvolution probability model. *Proc. IEEE SSP*, 2007.

[28] M. R. Gupta, H. S. Anderson, and Y. Chen. Joint deconvolution and classification for signals with multipath. *Proc. IEEE ICASSP*, 2007.

[29] B. Haasdonk and H. Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine Learning*, 68:35–61, 2007.

[30] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, 2001.

[31] R. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[32] L. Isserlis. On a formula for the product-mean coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12:134–139, 1918.

[33] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[34] K. Jamieson, M. R. Gupta, E. Swanson, and H. S. Anderson. Training a support vector machine to classify signals in a real environment given clean training data. *Proc. IEEE ICASSP*, 2010.

[35] T. Jebara. *Machine Learning: Discriminative and Generative.* Kluwer Academic Publishers, 2004.

[36] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844, 2004.

[37] J. N. Kapur. *Maximum-entropy models in science and engineering.* Wiley Eastern Limited, 1993.

[38] D. H. Kil and F. B. Shin. *Pattern Recognition and Prediction with Applications to Signal Characterization.* AIP Press, Woodbury, New York, 1996.

[39] L. E. Kinsler and A. R. Frey. *Fundamentals of Acoustics.* John Wiley & Sons, Inc., 2nd edition, 1962.

[40] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems.* Springer, 1996.

[41] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[42] Edmund Y. Lam and Joseph W. Goodman. Iterative statistical approach to blind image deconvolution. *J. Opt. Soc. Am.*, 17(7):1177–1184, July 2000.

[43] H. Liu, P. Runkle, and L. Carin. Classification of distant targets situated near channel bottoms. *J. Acoust. Soc. Am.*, 115(3):1185–1197, 2004.

[44] A. J. Llorens, T. L. Philip, I. W. Schurman, and C. R. Lorenz. Enhancing passive automation performance using an acoustic propagation simulation. *J. Acoust. Soc. Am*, 125(4):2577, April 2009.

[45] P. Loughlin and L. Cohen. Moment features invariant to dispersion. *Proc. SPIE*, 5426(235), 2004.

[46] David K. Mellinger and Christopher W. Clark. Mobysound: A reference archive for studying automatic recognition of marine mammal sounds. *J. App. Acous.*, 67(11-12):1226–1242, Nov-Dec 2006.

[47] T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

[48] F. D. Neeser and J. L. Massey. Proper complex random processes with applications to information theory. *IEEE Trans. Info. Theory*, 39(4), July 1993.

[49] G. Okopal and P. Loughlin. Feature extraction for classification of signals propagating in channels with dispersion and dissipation. *Proc. SPIE, Aut. Target Rec.*, 6566, 2007.

[50] G. Okopal, P. Loughlin, and L. Cohen. Dispersion-invariant features for classification. *J. Acoust. Soc. Am*, 123(2):832–841, February 2008.

[51] G. Okopal and P. J. Loughlin. Propagation-invariant classification of signals in channels with dispersion and damping. *OCEANS*, 2007.

[52] A. P. Petropulu and C. L. Nikias. Blind deconvolution using signal reconstruction from partial higher order cepstral information. *IEEE Trans. Signal Processing*, 41(6), June 1993.

[53] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, 2009.

[54] B. Raj and R. M. Stern. Missing-feature approaches in speech recognition. *Sig. Proc. Magazine*, (5):101–116.

[55] M. J. Roan, M. R. Gramann, J. G. Erling, and L. H. Sibuld. Blind deconvolution applied to acoustical systems identification with supporting experimental results. *J. Acoust. Soc. Am.*, 114(4):1988–1996, 2003.

[56] B. Schölkopf, C. Burgess, and V. Vapnik. Incorporating invariances in support vector learning machines. *Int'l Conf. Neural Networks*, pages 47–52, 1996.

[57] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. *Adv. in Neural Inf. Proc. Sys.*, 1998.

[58] S. Senmoto and D.G. Childers. Signal resolution via digital inverse filtering. *IEEE Trans. Aer. & Elec. Sys.*, 8(5):644–640, 1972.

[59] F. B. Shin, D. H. Kil, and R. Wayland. IER clutter reduction in shallow water. *Proc. IEEE ICASSP*, pages 3149–3152, 1996.

[60] S. Srivastava, M. R. Gupta, and B. A. Frigyik. Bayesian quadratic discriminant analysis. *J. Mach. Learn. Res.*, 8:1277–1305, 2007.

[61] H. Stark and J. W. Woods. *Probability and Random Proceses with Applications to Signal Processing.* Prentice Hall, 3rd edition, 2002.

[62] D. J. Strausberger, E. D. Garber, N. F. Chamberlain, and E. K. Walton. Modeling and performance of HF/OTH radar target classification systems. *IEEE Trans. Aer. & Elec. Sys.*, 28(2):396–403, 1992.

[63] D. W. Tufts, I.P. Kirsteins, R. J. Vaccaro, C. S. Ramalingam, and A. Shah. Improvements in signal processing for time-varying multipath propagation environments. *Proc. OCEANS*, 1:586–593, 1991.

[64] R. J. Urick. *Principles of Underwater Sound.* McGraw-Hill, 3rd edition, 1983.

[65] R. A. Wiggins. Minimum entropy deconvolution. *Geoexploration*, page 2135, 1978.

[66] R. W. Yeung. *A First Course in Information Theory.* Springer, 2002.

[67] Q. Zhu and B. Steinberg. Correction of multipath interference using clean and spatial location diversity. *Proc. IEEE Ultrasonics Symp.*, pages 1367–1370, 1995.

[68] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Appendix A

USEFUL IDENTITIES

The following identities are used frequently in this thesis.

A.1 Convolution and Hadamard Product of Vectors

For any vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^d$, scalar $\beta \in \mathbb{C}$,

$$(\mathbf{a} * \mathbf{b})(\mathbf{a} * \mathbf{b})^T = \left(\mathbf{a}\mathbf{a}^T\right) * * \left(\mathbf{b}\mathbf{b}^T\right) \tag{A.1}$$

$$(\mathbf{a} \cdot \mathbf{b})(\mathbf{a} \cdot \mathbf{b})^T = \left(\mathbf{a}\mathbf{a}^T\right) \cdot \left(\mathbf{b}\mathbf{b}^T\right) \tag{A.2}$$

$$\mathbf{a}\mathbf{b}^T \cdot (\beta I) = \beta \operatorname{diag}(\mathbf{a} \cdot \mathbf{b}), \tag{A.3}$$

which can be verified by writing the relationships in summation form.

For $A \in \mathbb{C}^{n \times n}$, $\mathbf{h} \in \mathbb{C}^p$ and $H = \text{convmtx}(\mathbf{h}) \in \mathbb{C}^{m \times n}$, where $m = n + p - 1$ and

$$\text{convmtx}(\mathbf{h}) = \begin{bmatrix} h_1 & \cdots & h_p & 0 & \cdots & 0 \\ 0 & h_1 & \cdots & h_p & \cdots & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & h_1 & \cdots & h_p \end{bmatrix},$$

it can also be shown that

$$HAH^T = A * *\mathbf{h}\mathbf{h}^T.$$

A.2 Convolution and Hadamard Product of Tensors

The properties for vectors in (A.1) and (A.2) can be generalized for tensors. Let A and B be d-order tensors with $A \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ and $B \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, and let $*^d$ denote the d-dimensional discrete convolution sum. Then, it is simple to verify (for example, using Einstein summation convention), that

$$\left(A *^d B\right) \circ \left(A *^d B\right) = (A \circ A) *^{2d} (B \circ B) \tag{A.4}$$

$$(A \cdot B) \circ (A \cdot B) = (A \circ A) \cdot (B \circ B),$$

where the tensor outer product $A \circ B$ is a $2d$-order tensor with dimensions $m_1 \times n_1 \times \cdots \times m_d \times n_d$.

A.3 Proper White Gaussian RVs

Let $\mathbf{V} \in \mathbb{C}^d$ be a complex Gaussian random vector that is zero mean, white, and proper. By definition of proper complex vectors [48],

$$\mathrm{E}\left[\mathbf{V}\mathbf{V}^H\right] = \mathrm{E}\left[\mathbf{V}^*\mathbf{V}^T\right] = \sigma_v^2 I \tag{A.5}$$

$$\mathrm{E}\left[\mathbf{V}\mathbf{V}^T\right] = 0. \tag{A.6}$$

Since $\mathbf{V}$ is zero-mean and Gaussian, the third moment is zero, which implies

$$\begin{aligned} \mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,\mathbf{V}^T\right] &= \mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,\mathbf{V}^H\right] \\ = \mathrm{E}\left[\mathbf{V}\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] &= \mathrm{E}\left[\mathbf{V}^*\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] = 0. \end{aligned} \tag{A.7}$$

Then, using Isslerlis' Gaussian moment theorem [32], it is straightforward to show that

$$\mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] = \sigma_v^4 I + \sigma_v^4 \mathbf{1}\mathbf{1}^T, \tag{A.8}$$

where $\mathbf{1}\mathbf{1}^T$ is a matrix of all ones.

A.4 Product of Gaussians Identity

Lastly, for $\mathbf{x}, \mathbf{a} \in \mathbb{R}^n$, $A \in \mathbb{S}^n_{++}$, $P \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $B \in \mathbb{S}^m_{++}$,

$$\mathcal{N}(\mathbf{x};\mathbf{a},A)\mathcal{N}(P\mathbf{x};\mathbf{b},B) = \mathcal{N}\left(\mathbf{b};P\mathbf{a},B+PAP^T\right)\mathcal{N}(\mathbf{x};\mathbf{c},C), \tag{A.9}$$

where $c = C\left(A^{-1}\mathbf{a} + PB^{-1}\mathbf{b}\right)$ and $C = \left(A^{-1} + P^TB^{-1}P\right)^{-1}$.

Appendix B

DERIVATIONS

B.1 Derivation of Covariance of $\mathbf{U}_z$

Let $\mathbf{X}^f$, $\mathbf{H}^f$ and $\mathbf{W}^f$ be mutually independent random vectors in the subband energy relationship in (4.1), and let $\mathbf{W}^f$ be a complex Gaussian vector that is zxero mean, white, and proper. Then,

$$
\begin{aligned}
\mathrm{Cov}\left[\mathbf{U}_z\right] = {} & \mathrm{Cov}\left[\mathbf{U}_h \cdot \mathbf{U}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right] \\
= {} & \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] + \mathrm{E}\left[\mathbf{U}_w \mathbf{U}_w^T\right] \\
& + 4\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.1)} \\
& + \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\mathbf{U}_w^T\right] + \mathrm{E}\left[\mathbf{U}_w\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] && \text{(B.2)} \\
& + 2\,\mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.3)} \\
& + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] && \text{(B.4)} \\
& + 2\,\mathrm{E}\left[\mathbf{U}_w\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.5)} \\
& + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\mathbf{U}_w^T\right] && \text{(B.6)} \\
& - \left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T - \bar{\mathbf{U}}_w \bar{\mathbf{U}}_w^T \\
& - \bar{\mathbf{U}}_w\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T - \left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\bar{\mathbf{U}}_w^T, && \text{(B.7)}
\end{aligned}
$$

where additional terms involving $\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right]$ are zero because $\mathbf{W}^f$ is zero mean and independent of $\mathbf{X}^f$ and $\mathbf{H}^f$. Line (B.2) cancels with (B.7). Lines (B.3) and (B.4) are zero since $\mathbf{W}^f$ is zero mean and uncorrelated with $\mathbf{X}^f$ and $\mathbf{H}^f$. Since $\mathbf{W}^f$ is proper and $\mathbf{U}_w = \mathbf{W}^f \cdot \mathbf{W}^{f*}$, lines (B.5) and (B.6) are zero by property (A.7). By expanding

$\mathrm{Re}\{\mathbf{a}\} = \frac{1}{2}(\mathbf{a} + \mathbf{a}^*)$ and multiplying, line (B.1) can be rewritten as

$$\mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] \tag{B.8}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] \tag{B.9}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] \tag{B.10}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] \tag{B.11}$$

$$= \mathrm{E}\left[\left(\mathbf{X}^f \mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f \mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*} \mathbf{W}^{fT}\right)\right] \tag{B.12}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^{f*} \mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*} \mathbf{H}^{fT}\right) \cdot \left(\mathbf{W}^f \mathbf{W}^{fH}\right)\right]. \tag{B.13}$$

where properties (A.2) and (A.6) can be used to verify that lines (B.8) and (B.11) are zero. Using (A.2), lines (B.9) and (B.10) become (B.12) and (B.13), respectively. This yields

$$\mathrm{Cov}[\ \mathbf{U}_z] = \mathrm{E}\left[(\mathbf{U}_h \cdot \mathbf{U}_x)(\mathbf{U}_h \cdot \mathbf{U}_x)^T\right] \tag{B.14}$$

$$-\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T \tag{B.15}$$

$$+\ \mathrm{E}\left[\mathbf{U}_w \mathbf{U}_w^T\right] - \bar{\mathbf{U}}_w \bar{\mathbf{U}}_w^T \tag{B.16}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^f \mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f \mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*} \mathbf{W}^{fT}\right)\right] \tag{B.17}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^{f*} \mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*} \mathbf{H}^{fT}\right) \cdot \left(\mathbf{W}^f \mathbf{W}^{fH}\right)\right] \tag{B.18}$$

$$=\ \mathrm{E}\left[\mathbf{U}_h \mathbf{U}_h^T \cdot \mathbf{U}_x \mathbf{U}_x^T\right] - \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T \cdot \bar{\mathbf{U}}_x \bar{\mathbf{U}}_x^T \tag{B.19}$$

$$+\ \mathrm{E}\left[\mathbf{U}_w \mathbf{U}_w^T\right] - \bar{\mathbf{U}}_w \bar{\mathbf{U}}_w^T \tag{B.20}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^f \mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f \mathbf{H}^{fH}\right)\right] \cdot \sigma_w^2 I \tag{B.21}$$

$$+\ \mathrm{E}\left[\left(\mathbf{X}^{f*} \mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*} \mathbf{H}^{fT}\right)\right] \cdot \sigma_w^2 I, \tag{B.22}$$

where property (A.2) was used to rewrite (B.14) and (B.15) as (B.19). Then, use (A.5) to simplify (B.17) and (B.18) as, respectively, (B.21) and (B.22). In (B.21) and (B.22), $\mathrm{E}\left[\left(\mathbf{X}^f \mathbf{X}^{fH}\right)\right] \cdot \sigma_w^2 I = \sigma_w^2 \,\mathrm{diag}\left(\mathrm{E}\left[\mathbf{X}^f \cdot \mathbf{X}^{f*}\right]\right) = \mathrm{diag}\left(\bar{\mathbf{U}}_x\right)$ by (A.3) and by definition of $\mathbf{U}_x$ (similarly for terms involving $\mathbf{H}^f$). Thus, (B.21) and (B.22) simplify to $2\sigma_w^2 \,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)$. Applying (A.8) to $\mathrm{E}\left[\mathbf{U}_w \mathbf{U}_w^T\right]$, and recalling that $\bar{\mathbf{U}}_w = \sigma_w^2 \mathbf{1}$, (B.20) reduces to $\sigma_w^4 I$. Finally, $\mathrm{E}\left[\mathbf{U}_h \mathbf{U}_h^T \cdot \mathbf{U}_x \mathbf{U}_x^T\right] - \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T \cdot \bar{\mathbf{U}}_x \bar{\mathbf{U}}_x^T = \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_x} + \Sigma_{u_h} \cdot$

$\bar{\mathbf{U}}_x \bar{\mathbf{U}}_x^T$, so that be collecting terms, we have

$$\mathrm{Cov}\left[\mathbf{U}_z\right] = \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_x} + \Sigma_{u_h} \cdot \bar{\mathbf{U}}_x \bar{\mathbf{U}}_x^T + \sigma_w^4 I + 2\sigma_w^2 \,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right). \tag{B.23}$$

B.2 Derivation of Expected RBF for Dependent $\mathbf{U}_{z_i}$ and $\mathbf{U}_{z_j}$

In the following, we assume that conditioning on $\mathbf{u}_{x_i}, \mathbf{u}_{x_j}$ is implicit, that is $p(\mathbf{u}_{z_i}, \mathbf{u}_{z_j}) = p(\mathbf{u}_{z_i}, \mathbf{u}_{z_j} | \mathbf{u}_{x_i}, \mathbf{u}_{x_j})$.

Model $p(\mathbf{u}_{z_i}, \mathbf{u}_{z_j}) = \mathcal{N}\left(\begin{bmatrix} \bar{\mathbf{U}}_{z_i} \\ \bar{\mathbf{U}}_{z_j} \end{bmatrix}, \begin{bmatrix} \Sigma_{u_{z_i}} & \Gamma^T \\ \Gamma & \Sigma_{u_{z_j}} \end{bmatrix}\right)$, where $\Sigma_{u_{z_i}}$ is given in (B.23) by substituting $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$ and $\Sigma_{u_x} = 0$, since we condition on $\mathbf{U}_x = \mathbf{u}_{x_i}$:

$$\Sigma_{u_{z_i}} = \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2 \,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right).$$

The cross-covariance term Γ is solved for in the following manner:

$$\begin{aligned}
\Gamma =& \,\mathrm{E}\left[\mathbf{U}_{z_i} \mathbf{U}_{z_j}^T\right] - \bar{\mathbf{U}}_{z_i} \bar{\mathbf{U}}_{z_j}^T \\
=& \,\mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{u}_{x_i} + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right)\right. \\
& \left.\left(\mathbf{U}_h \cdot \mathbf{u}_{x_j} + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right)^T\right] \\
& - \bar{\mathbf{U}}_{z_i} \bar{\mathbf{U}}_{z_j}^T \\
=& \,\mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{u}_{x_i}\right)\left(\mathbf{U}_h \cdot \mathbf{u}_{x_j}\right)^T\right] + \mathrm{E}\left[\mathbf{U}_w \mathbf{U}_w^T\right] && \text{(B.24)} \\
& + 4\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\} \mathrm{Re}\left\{\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.25)} \\
& + \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{u}_{x_i}\right) \mathbf{U}_w^T\right] + \mathrm{E}\left[\mathbf{U}_w \left(\mathbf{U}_h \cdot \mathbf{u}_{x_j}\right)^T\right] && \text{(B.26)} \\
& + 2\,\mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{u}_{x_i}\right) \mathrm{Re}\left\{\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.27)} \\
& + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\} \left(\mathbf{U}_h \cdot \mathbf{u}_{x_j}\right)^T\right] && \text{(B.28)} \\
& + 2\,\mathrm{E}\left[\mathbf{U}_w \,\mathrm{Re}\left\{\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && \text{(B.29)} \\
& + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\} \mathbf{U}_w^T\right] && \text{(B.30)} \\
& - \left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right)\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_j}\right)^T - \bar{\mathbf{U}}_w \bar{\mathbf{U}}_w^T && \text{(B.31)} \\
& - \bar{\mathbf{U}}_w \left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_j}\right)^T - \left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right) \bar{\mathbf{U}}_w^T, && \text{(B.32)}
\end{aligned}$$

where additional terms involving $\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right]$ are zero because $\mathbf{W}^f$ is zero mean and independent of $\mathbf{H}^f$. Lines (B.29) and (B.30) are zero by (A.7), and lines (B.27) and (B.28) are zero since $\mathbf{W}^f$ is zero mean. Further, line (B.26) cancels with (B.32). Line (B.25) can be expanded using $\mathrm{Re}\{a\} = \frac{1}{2}(\mathbf{a} + \mathbf{a}^*)$ to be

$$\mathrm{E}\left[\left(\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] \tag{B.33}$$

$$+\ \mathrm{E}\left[\left(\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] \tag{B.34}$$

$$+\ \mathrm{E}\left[\left(\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] \tag{B.35}$$

$$+\ \mathrm{E}\left[\left(\mathbf{x}_i^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{x}_j^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] \tag{B.36}$$

$$= \mathrm{E}\left[\left(\mathbf{x}_i^f \mathbf{x}_j^{fH}\right) \cdot \left(\mathbf{H}^f \mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*} \mathbf{W}^{fT}\right)\right] \tag{B.37}$$

$$+\ \mathrm{E}\left[\left(\mathbf{x}_i^{f*} \mathbf{x}_j^{fT}\right) \cdot \left(\mathbf{H}^{f*} \mathbf{H}^{fT}\right) \cdot \left(\mathbf{W}^f \mathbf{W}^{fH}\right)\right]. \tag{B.38}$$

Combining (B.24), (B.25) and (B.31), and using the same logic as used to derive (B.23), we have

$$\begin{aligned}
\Gamma =\ & \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{u}_{x_i}\right)\left(\mathbf{U}_h \cdot \mathbf{u}_{x_j}\right)^T\right] - \left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right)\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_j}\right)^T \\
& + \mathrm{Cov}\left[\mathbf{U}_w \mathbf{U}_w^T\right] \\
& + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\left(\mathbf{x}_i^f \mathbf{x}_j^{fH}\right) \cdot \left(\mathbf{H}^f \mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*} \mathbf{W}^{fT}\right)\right\}\right] \\
& = \Sigma_h \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_j}^T + \sigma_w^4 I + 2\sigma_w^2\, \mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot\ \mathrm{Re}\left\{\mathbf{x}_i^f \cdot \mathbf{x}_j^{f*}\right\}\right).
\end{aligned}$$

Since $p(\mathbf{u}_{z_i}, \mathbf{u}_{z_j})$ is jointly Gaussian in $\mathbf{u}_{z_i}$ and $\mathbf{u}_{z_j}$, we can rewrite $p(\mathbf{u}_{z_i}, \mathbf{u}_{z_j}) = p(\mathbf{u}_{z_i}|\mathbf{u}_{z_j})p(\mathbf{u}_{z_j})$, where $p(\mathbf{u}_{z_j}) = \mathcal{N}\left(\mathbf{u}_{z_j}; \bar{\mathbf{U}}_{z_j}, \Sigma_{z_j}\right)$ and $p(\mathbf{u}_{z_i}|\mathbf{u}_{z_j}) = \mathcal{N}\left(\mathbf{u}_{z_i}; \bar{\mathbf{U}}_{z_i} + \Omega\left(\mathbf{u}_{z_j} - \bar{\mathbf{U}}_{z_j}\right), \Psi\right)$, where $\Omega = \Gamma^T \Sigma_{z_j}^{-1}$ and $\Psi = \Sigma_{z_i} - \Gamma^T \Sigma_{z_j}^{-1} \Gamma$. Since the Gaussian RBF kernel is given by $K(\mathbf{U}_{z_i}, \mathbf{U}_{z_j}) = \mathcal{N}\left(\mathbf{U}_{z_i}; \mathbf{U}_{z_j}, \gamma^{-1} I\right)$, then by successive

use of the product of Gaussians identity in (A.9),

$$
\begin{aligned}
&\mathrm{E}_{\mathbf{U}_{z_i},\mathbf{U}_{z_j}}\left[K(\mathbf{U}_{z_i},\mathbf{U}_{z_j})\right] \\
&= \iint \mathcal{N}\left(\mathbf{u}_{z_i};\mathbf{u}_{z_j},\gamma^{-1}I\right) p(\mathbf{u}_{z_i},\mathbf{u}_{z_j})\, d\mathbf{u}_{z_i}\, d\mathbf{u}_{z_j} \\
&= \iint \mathcal{N}\left(\mathbf{u}_{z_i};\mathbf{u}_{z_j},\gamma^{-1}I\right) p(\mathbf{u}_{z_j}) p(\mathbf{u}_{z_i}|\mathbf{u}_{z_j})\, d\mathbf{u}_{z_i}\, d\mathbf{u}_{z_j} \\
&= \iint \mathcal{N}\left(\mathbf{u}_{z_i};\mathbf{u}_{z_j},\gamma^{-1}I\right) \mathcal{N}\left(\mathbf{u}_{z_j};\bar{\mathbf{U}}_{z_j},\Sigma_{z_j}\right) \\
&\qquad \mathcal{N}\left(\mathbf{u}_{z_i};\bar{\mathbf{U}}_{z_i}+\Omega\left(\mathbf{u}_{z_j}-\bar{\mathbf{U}}_{z_j}\right),\Psi\right)\, d\mathbf{u}_{z_i}\, d\mathbf{u}_{z_j} \\
&= \int \mathcal{N}\left(\mathbf{u}_{z_j};\bar{\mathbf{U}}_{z_i}+\Omega\left(\mathbf{u}_{z_j}-\bar{\mathbf{U}}_{z_j}\right),\gamma^{-1}I+\Psi\right) \\
&\qquad\qquad \mathcal{N}\left(\mathbf{u}_{z_j};\bar{\mathbf{U}}_{z_j},\Sigma_{z_j}\right)\, d\mathbf{u}_{z_j} \\
&= \int \mathcal{N}\left((I-\Omega)\,\mathbf{u}_{z_j};\bar{\mathbf{U}}_{z_i}-\Omega\bar{\mathbf{U}}_{z_j},\gamma^{-1}I+\Psi\right) \\
&\qquad\qquad \mathcal{N}\left(\mathbf{u}_{z_j};\bar{\mathbf{U}}_{z_j},\Sigma_{z_j}\right)\, d\mathbf{u}_{z_j} \\
&= \mathcal{N}\left((I-\Omega)\,\bar{\mathbf{U}}_{z_j};\bar{\mathbf{U}}_{z_i}-\Omega\bar{\mathbf{U}}_{z_j},\right. \\
&\qquad\qquad \left.(I-\Omega)\,\Sigma_{z_j}\,(I-\Omega)^T+\gamma^{-1}I+\Psi\right) \\
&= \mathcal{N}\left(\bar{\mathbf{U}}_{z_i};\bar{\mathbf{U}}_{z_j},(I-\Omega)\,\Sigma_{z_j}\,(I-\Omega)^T+\gamma^{-1}I+\Psi\right)
\end{aligned}
$$

Substituting in $\Omega = \Gamma^T\Sigma_{z_j}^{-1}$ and $\Psi = \Sigma_{z_i} - \Gamma^T\Sigma_{z_j}^{-1}\Gamma$, and cancelling terms yields

$$
\mathcal{N}\left(\bar{\mathbf{U}}_{z_i};\bar{\mathbf{U}}_{z_j},\Sigma_{z_i}+\Sigma_{z_j}-\left(\Gamma+\Gamma^T\right)+\gamma^{-1}I\right).
$$

VITA

Hyrum S. Anderson graduated with his Ph.D. in Electrical Engineering from the University of Washington in 2010. Previously, he was an associate staff member at MIT Lincoln Laboratory. He received his MS and BS degrees in Electrical Engineering from Brigham Young University, both in 2003, with emphases in remote sensing and signal processing. He took a two year leave of absence during his undergraduate degree to serve a voluntary two-year mission for his church, living in and around Moscow, Russia. He was raised in a large family in Meridian, Idaho.