ABSTRACT

Title of Document: DETECTION AND CLASSIFICATION OF NON-STATIONARY SIGNALS USING SPARSE REPRESENTATIONS IN ADAPTIVE DICTIONARIES

Daniela I. Moody, Doctor of Philosophy, 2012

Directed By: Professor & Chair, Patrick G. O'Shea
Department of Electrical and Computer Engineering

Automatic classification of non-stationary radio frequency (RF) signals is of particular interest in persistent surveillance and remote sensing applications. Such signals are often acquired in noisy, cluttered environments, and may be characterized by complex or unknown analytical models, making feature extraction and classification difficult. This thesis proposes an adaptive classification approach for poorly characterized targets and backgrounds based on sparse representations in non-analytical dictionaries learned from data. Conventional analytical orthogonal dictionaries, e.g., Short Time Fourier and Wavelet Transforms, can be suboptimal for

classification of non-stationary signals, as they provide a rigid tiling of the time-frequency space, and are not specifically designed for a particular signal class. They generally do not lead to sparse decompositions (i.e., with very few non-zero coefficients), and use in classification requires separate feature selection algorithms. Pursuit-type decompositions in analytical overcomplete (non-orthogonal) dictionaries yield sparse representations, by design, and work well for signals that are similar to the dictionary elements. The pursuit search, however, has a high computational cost, and the method can perform poorly in the presence of realistic noise and clutter. One such overcomplete analytical dictionary method is also analyzed in this thesis for comparative purposes. The main thrust of the thesis is learning discriminative RF dictionaries directly from data, without relying on analytical constraints or additional knowledge about the signal characteristics. A pursuit search is used over the learned dictionaries to generate sparse classification features in order to identify time windows that contain a target pulse. Two state-of-the-art dictionary learning methods are compared, the K-SVD algorithm and Hebbian learning, in terms of their classification performance as a function of dictionary training parameters. Additionally, a novel hybrid dictionary algorithm is introduced, demonstrating better performance and higher robustness to noise. The issue of dictionary dimensionality is explored and this thesis demonstrates that undercomplete learned dictionaries are suitable for non-stationary RF classification. Results on simulated data sets with varying background clutter and noise levels are presented. Lastly, unsupervised classification with undercomplete learned dictionaries is also demonstrated in satellite imagery analysis.

DETECTION AND CLASSIFICATION OF NON-STATIONARY SIGNALS

USING SPARSE REPRESENTATIONS IN ADAPTIVE DICTIONARIES

By

Daniela I. Moody

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor Patrick G. O'Shea, Chair
Professor Ramalingam Chellappa
Professor Christopher C. Davis
Dr. Steven P. Brumby
Professor Larry S. Davis

Dedication

To Nathan and Ethan.

Acknowledgements

I would like to express my gratitude to Dr. Steven Brumby for his inspiration and example, and for exposing me to the world of computational neuroscience. His mentoring has shaped my outlook and approach to research and his confidence-boosting methods are unparalleled. Also, Professor Patrick O'Shea has been a tremendous thesis advisor and provided guidance through several research transitions. He shared many valuable insights, both personal and academic in nature, and I am honored to have him as a life coach.

Dr. Bruce Carlsten was the first to welcome me as a graduate student at Los Alamos National Laboratory (LANL) and to give me the outdoor experience of the Southwest. Dr. John Galbraith, my first technical project lead at LANL, introduced me to experimental RF engineering and its practical challenges. I would like to thank Dr. Norma Pawley and Dr. Kary Myers for their daily support, guidance, and expert advice during my second technical project at LANL. Special thanks to Dr. James Theiler for providing invaluable machine learning theory insight, and to Dr. Brad Cooke for sharing his time-domain RF processing expertise. Dr. Rick Chartrand, Dr. Brendt Wohlberg, and Dr. Amy Galbraith provided very useful discussion on the practicalities of applied math theory. Thanks also to Dr. Joel Rowland and Dr. Chandana Gangodagamage who provided arctic hydrology and vegetation domain expert input. Support for this research comes from the National Nuclear Security

Administration and the Los Alamos National Laboratory Directed Research and Development (LDRD) Office (LA-UR-12-21994).

I wish to thank Professors Ramalingam Chellappa, Christopher Davis, and Larry Davis for serving on my dissertation exam committee and providing many good ideas for my research. I am especially grateful to Professor Chellappa for mentoring me as a Masters student and giving me the first taste of pattern recognition.

Most importantly, I would also like to thank my family for their lifelong unwavering support and example, and especially my little son Ethan for giving me something to laugh about daily. And most of all, I must express immeasurable gratitude to my husband, colleague, and best friend Nathan for his indelible support of my dreams.

Table of Contents

List of Tables

List of Figures

1. Introduction

Detection and analysis of transitory electromagnetic (EM) signatures is important for persistent surveillance applications and remote sensing applications (e.g., lightning research). Current detection and classification systems are mostly tailored for stationary signals, and as such are likely to misinterpret transients as noise. In contrast, this thesis will focus on automatic classification of signals with high pulse-to-pulse variability. Such EM signals can exhibit both discrete and continuous dynamical behavior, e.g., trains of intermittent frequency-hopping or chirping pulses, combined with continuous time-varying emissions during a single pulse. The EM-generating process may persist over a wide range of time scales, and usually occurs in the presence of additive white noise and structured clutter, including emissions from similar sources. Robust detection and discrimination methods are therefore essential. Detection of such nonstationary, poorly characterized target signals against a complex, nonstationary background presents challenges for standard detection and classification approaches.

Extracting classification features of a radiofrequency (RF) signal typically relies on knowledge of the application domain in order to find feature vectors unique to a signal class and robust to background noise. Conventional localized data representations using fixed orthonormal dictionaries, such as a short-time Fourier basis [1] or a Best Orthonormal Basis [2] selected from a wavelet packet decomposition, can be suitable for analyzing some types of signals, but not others.

Successful classification of underwater acoustic transients using Daubechies wavelets [3] is a good demonstration of the advantage of wavelet analysis in the presence of white noise background. In contrast, using an optimized wavelet representation for space-based RF transient signal classification in the presence of a more complex background [4] has produced unsatisfactory results.

Fixed orthonormal (or complete) dictionaries do not usually lead to sparse decompositions for all types of signals, and require separate feature selection algorithms, resulting in additional computational overhead. The feature vector can be very sparse for one category of signal (e.g., constant frequency emitter using a Fourier basis), but dense for another (e.g., chirped pulse using a Fourier basis). One alternative is to employ a carefully chosen, redundant (or overcomplete) dictionary, from which we may be able to obtain sparse representations of data using a matching pursuit [5]. An example of an overcomplete dictionary with higher representation flexibility for RF signals is the chirped Gabor wavelet dictionary [6], which can be used in conjunction with the fast-ridge pursuit of [7], and will be further explored in Chapter 3. The resulting dictionary elements (also called atoms) can represent both pulses and CW signals in very few atoms from the dictionary [8].

A fixed dictionary of parameterized, closed-form atoms, whether complete or overcomplete, requires assumptions about the signal data which are not realistic in most applications. Learned or adaptive dictionaries avoid this constraint and lead to new methods described later in this work (Chapters 4 and 5). Initially introduced by Olshausen and Field [9] for modeling the mammalian visual cortex, the idea of

learning a dictionary directly from data has gained momentum in the image processing field. Coding theory and biologically inspired algorithms have been explored [9-12] to learn dictionaries for sparse representation of image edges and textures, panchromatic satellite imagery, and to model the visual cortex [13]. These learned dictionaries have led to significant improvements in image representation, classification, and image restoration. Recently, Mairal [14] has proposed learning overcomplete, non-parametric dictionaries optimized both for representation and classification of images.

In this thesis, the learned dictionary techniques are extended to RF data and results are presented demonstrating classification performance on a simulated data set. Both undercomplete and overcomplete dictionaries with respect to the dimensions of the input vectors are investigated. The goal is to identify the presence and capture the dynamic behavior of a chirped pulse target emitter while remaining robust to varying levels of background clutter and noise; that is, analysis time windows must be classified according to whether they contain a target pulse (ON) or not (OFF). A Hebbian learning algorithm is examined and compared to the K-SVD algorithm [10] in terms of how their respective learned elements (i.e., RF features) perform in classification, as a function of learning iterations and dictionary size. After building dictionaries of RF features with the two methods, their classification performance is compared using Skretting and Husøy's minimum residual (MR) classifier, originally introduced for texture classification [15]. The research focus is understanding the learning process and, more specifically, learning the discriminating RF features. The

quality of signal reconstruction is not a priority, although it will indirectly be evaluated in the course of the research. Rather, the performance metric of choice is *classification accuracy*. Much of the work in this dissertation has been included in [16-18]. Secondly, although the primary dissertation work is focused in the RF domain, the findings are used to generalize the undercomplete learned dictionary approach and extend it to unsupervised classification in multispectral imagery [19, 20].

1.1 Simulation environment

The majority of the work in this thesis is evaluated on representative synthetic RF data. The set of test conditions for the synthetic data was chosen to match the parameters (e.g., sampling rate, time resolution, frequency resolution) and challenges (e.g., abundance and magnitude of clutter) of actual measurements. The simulated data set consists of a target signal that operates intermittently (alternating ON/OFF states), emitting linearly chirped pulses at a base frequency of 220 kHz and pulse duration equal to 5 ms. This is a signal in the low range of RF, but methodologies of classifying signals are frequency scalable, meaning the mathematical treatment is the same (but more samples and computation are required). Given that many real-world classification problems will be in the UHF (400MHz) to microwave regions (4 GHz), some of the methods proposed in this thesis remain practical in terms of computation time for analysis of these real world signals. To aid with computation, high frequency

signals can be down-converted to lower frequency by mixing the data signal with a local oscillator.

Three target amplitudes are considered: high, mid, and low, with corresponding SNR 3:1, 1:1, and 0.3:1. The background is modeled as a superposition of additive white Gaussian noise and clutter, consisting of several continuous wave (CW) signals and a competing linear chirp pulse emitter. This competing emitter operates at a base frequency close to or within the target spectrum region, and has characteristic time scales for pulse duration and pulse spacing similar to the target emitter. Three different pulsed emitter cases are considered and labeled below for reference throughout the following chapters, each simulating increasingly complex data:

- Linear chirp pulse emitter, start frequency far from the end-of-chirp target frequency ***(Far CP)***
- Constant frequency pulse emitter, within the target spectrum ***(Flat CP)***
- Linear chirp pulse emitter, with the same target frequency and different chirp rate ***(Chirped CP)***

Figure 1.1 shows a sample waveform from Chirped CP data. The CW and chirped clutter signals have amplitudes equal to or greater than that of the target, and they span a frequency range of 30 kHz – 490 kHz, as illustrated in Figure 1.2. The phases of each pulse emitter are uncorrelated; the clutter therefore is non-stationary

with respect to the target. The spectrograms in Figure 1.2 show the relative complexity of the resulting timeseries in each of the three clutter cases, and for each SNR scenario. Table 1.1 summarizes the characteristics of the simulated data sets, where the amplitude values are in units relative to the noise variance.

The modeled data recording system operates at a sampling rate of 1 MHz, and buffers 0.5 s of data at a time (i.e., output timeseries are $5x10^5$ samples long). The data is processed using data analysis windows of length N=512 samples (0.5 ms of recording) with overlap of 256 samples, and the goal is to correctly identify the operational state of the target in each window. A window is classified as an "ON-window" if the target pulse is present for the entire window. A window is labeled as an "OFF-window" if the target pulse is completely absent. Windows containing target signal of partial duration are ignored. This window-level classification can then be used in a hierarchical, dynamic process analysis system for large time-scale "target mode" classification similar to the one detailed in [8].

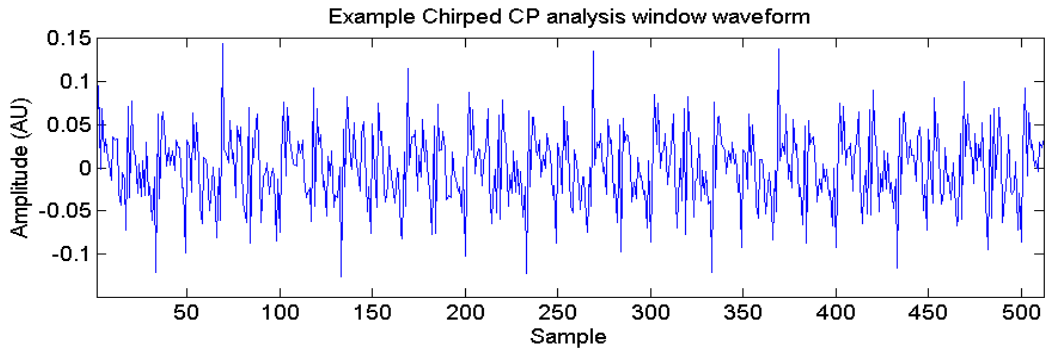


Figure 1.1: Sample signal amplitude (arbitrary units) vs. sample length of an analysis window of Chirped CP data with SNR 3:1. Visual inspection of time-domain data is uninformative with respect to the signal components; time series appears as noise throughout the analysis window.

Table 1.1: Simulation environment characteristics

Signal component	Parameters		
	Frequency	*Chirp rate*	*Amplitude*
Target	220 kHz	15.5 MHz/s	{3, 1, 0.3}*
CW clutter	{30 kHz, 60 kHz, 120 kHz, 210 kHz, 310 kHz, 370 kHz, 430 kHz, 490 kHz}	0	{ 3, 5, 6, 4, 3, 5, 4, 6}
Far CP	320 kHz	32 MHz/s	1
Flat CP	260 kHz	0	3
Chirped CP	220 kHz	32 MHz/s	3
Gaussian noise	N/A	N/A	1

* Only relative target amplitude changes for the SNR 3:1, 1:1, and 0.3:1 data; the noise and clutter levels remain identical.

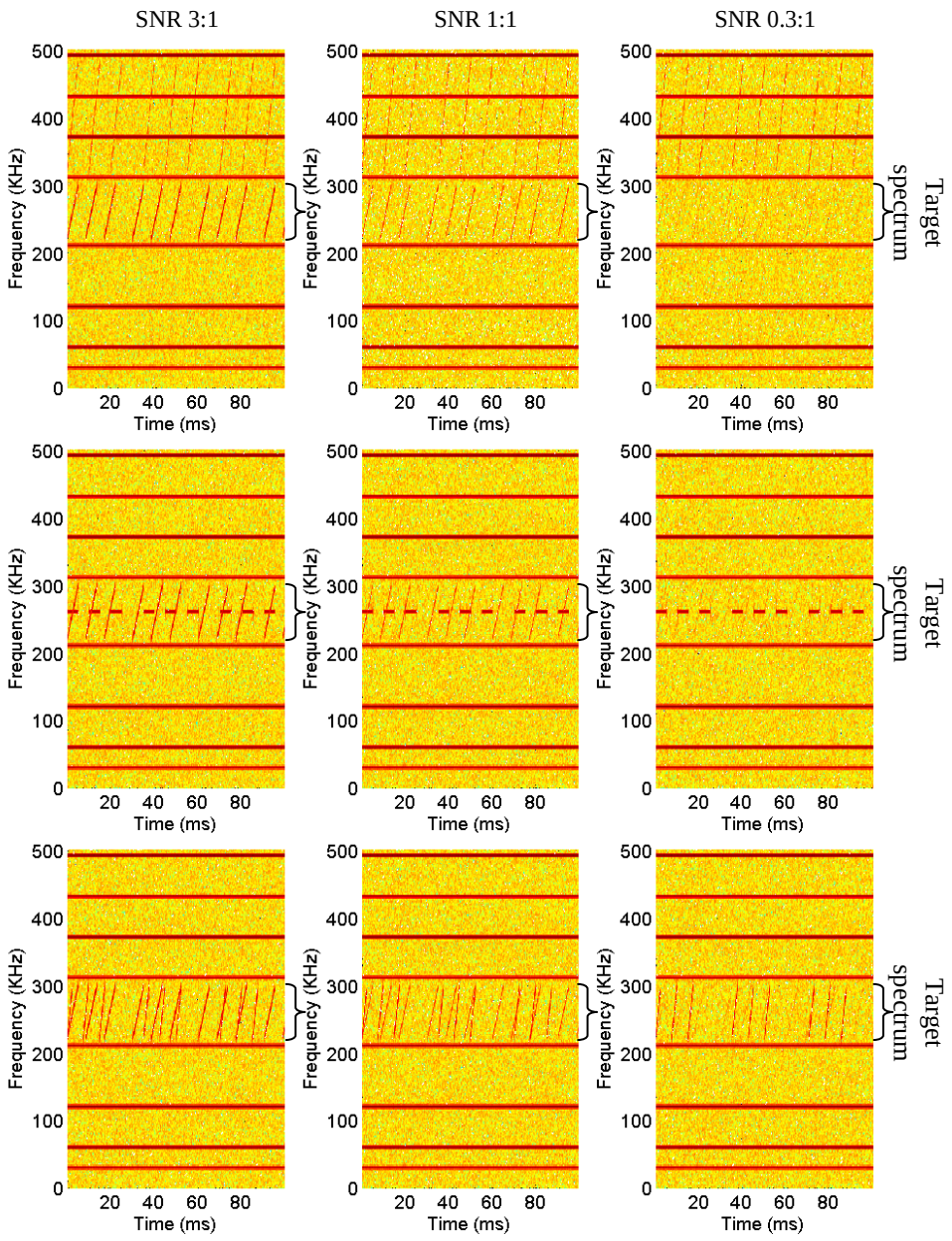


Figure 1.2: Spectrograms illustrating the signal, noise, and clutter characteristics of Far CP (top row), Flat CP (middle row), and Chirped CP (bottom row) data, for each noise scenario: SNR 3:1 (left column), SNR 1:1 (middle column), and SNR 0.3:1 (right column). The target spectrum is marked with a black bracket. For each SNR case the target is made "quieter" by decreasing its relative strength, making its characteristics increasingly difficult to discern (left to right) in the spectrograms.

1.2 Summary of methods

In traditional signal processing, it is common to use orthonormal transforms on the input data to facilitate feature extraction or component analysis. These transforms, or basis, are by definition orthogonal, complete (i.e., number of elements is equal to natural input dimensionality), and have a projection. That is, the coefficients can be obtained by direct projection onto the basis, and the input can be re-synthesized with perfect reconstruction from these coefficients. In general, the vectors of coefficients are not always ***sparse***, where sparse is taken to mean that only a small number of coefficients are non-zero. In recent years, the notion of basis has been extended to that of ***dictionaries*** by relaxing the orthogonality and completeness constraints. Dictionaries no longer provide the option for a direct projection and alternative search methods must be employed to obtain the coefficient vector. While there can be many non-unique approximations of an input, when the vector of coefficients is enforced to be sparse, the decomposition is unique. The reconstruction is now in general only approximate, but it is obtained from few coefficients.

Two main thrust areas have been identified in current literature as potentially useful for the RF classification problem at hand. Fundamentally, both rely on the notion of dictionary, and will be introduced in greater detail in Chapter 2. The first one relies on using adaptive sparse representations in redundant, parametric dictionaries (i.e., with elements generated by an analytical or closed-form function). The second one relies on using sparse representations in adaptive learned dictionaries (i.e., with elements learned directly from data). Both of these approaches will be

extended to RF classification and evaluated in terms of their potential for generating discriminative features and classifying simulated RF data. The work presented in this thesis is the first extension of learned dictionary techniques to RF signal processing, to the author's best knowledge (derived from literature searches and expert consultation). Also novel in this thesis is the use of classification accuracy, instead of reconstruction accuracy, as the primary performance metric for such learned dictionaries. An example of one parametric dictionary case will be shown in which good reconstructive ability does not imply good classification potential. For learned dictionaries, it will also be shown that good classification can be obtained without having perfect reconstruction. In addition to classification accuracy, other factors in the evaluation include computational complexity and robustness to noise.

1.3 Thesis outline

The layout of the thesis is as follows: Chapter 2 offers a comprehensive literature review on RF classification, sparse representations, and dictionary methods, both parametric and learned. Chapter 3 focuses on representations using overcomplete parametric dictionaries. Two dictionary learning methods and their parameters are explored in Chapter 4 for the Far CP data. A new, unpublished approach of learning hybrid dictionaries (i.e., dictionaries that use a hierarchical learning method that combines different algorithms) for better performance is introduced and detailed toward the end of this chapter. Chapter 5 demonstrates RF classification for the more complex Flat CP and Chirped CP datasets, using sparse

representations in undercomplete learned dictionaries, which is a key novel component of this thesis. In Chapter 6, the methodology and framework of learning undercomplete dictionaries for classification is showcased in a very different application: multispectral satellite imagery. The specific topic of Chapter 6 is unsupervised classification (i.e., classification that cannot be iteratively checked for accuracy during the learning process) of land cover in the Arctic. Chapter 7 concludes with discussion of results and future directions. The focus of the last two chapters is to demonstrate that the methods developed as part of this thesis do indeed address real-world problems, and show promise for solving issues in other application areas not specifically covered in this work.

2. Background and Fundamentals

The general task of classification or pattern recognition is accomplished through several sequential steps, as illustrated in Figure 2.1 [21]. The *sensing step* involves hardware devices (e.g., antenna, analog to digital converter) which convert raw RF input into sampled signal data. Given that simulated data is used in this thesis, the sensing step is not explicitly addressed. The *segmentation* consists of separating signals from the background or from other signals of interest, and can be a complex problem depending on the particular pattern recognition application. For the simulated RF data under consideration, segmentation can be either target signal versus everything else, or ON versus OFF windows. The former will be the choice in Chapter 3, and the latter will be the choice in Chapters 4 and 5.

The next two steps, *feature extraction* and *classification*, are interdependent and a conceptual boundary between the two is arbitrary [21]. The task of feature extraction consists of finding discriminating features (or feature vectors) that are invariant to certain input changes (e.g., instantaneous frequency of chirping target pulse in the current analysis window), and robust to changes in overall amplitude of target with respect to that of noise and clutter. Following the feature extraction, the classifier is the specific algorithm that uses the feature vectors to assign a signal to a class or category. In the extreme cases, an ideal feature extractor would only need a very simple classifier, and conversely an ideal classifier would not need sophisticated

feature extraction. The classifier outputs a *decision* (i.e., signal class), which in this thesis is evaluated based on classification accuracy.

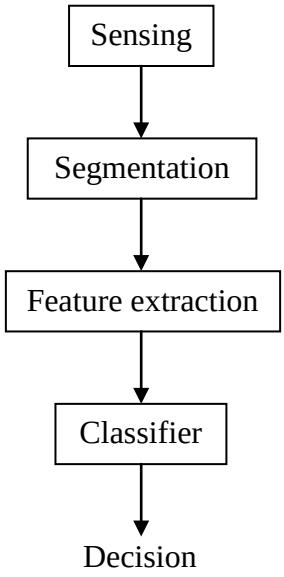


Figure 2.1: Steps in a general classification process. The sensing encompasses devices necessary to convert low level RF input into digital sampled signal data. Segmentation separates signal components from each other or from the background. Feature extraction finds robust, discriminative features for each signal class of interest; the classifier is an algorithm that uses these features to come to a final decision regarding the class of signal. In this thesis, the decision is evaluated in terms of classification accuracy.

Automatic classification of (broadband) non-stationary RF signals is of particular interest in some applications (e.g., lightning research). Because such signals are often acquired in noisy, cluttered environments, and are characterized by complex or unknown analytical models, feature extraction and subsequent classification can be difficult. Extracting features typically requires good knowledge of the application domain in order to find feature vectors unique to a class and robust to background noise. One of the main underlying goals of this thesis is robust feature extraction from

non-stationary signals, and so the associated classifier is chosen to be relatively simple in order to allow performance to be driven primarily by the features.

Section 2.1 gives a broad overview of the various signal analysis methods that have conventionally been used for feature extraction, followed in Section 2.2 by a literature survey highlighting the specific use of one of these methods, wavelet analysis, in non-stationary signal processing. Section 2.3 presents published results on classification of transient one-dimensional signals, and summarizes the extensive literature available on general design of classifier algorithms. Section 2.4 motivates and introduces dictionary methods, which have been evolving during the past decade and will be the focus of this dissertation.

2.1 Feature extraction using conventional signal processing

A rich history, with roots in the classical numerical analysis techniques of the 17th century [1], has led to the present state-of-the-art methods in signal processing. This section summarizes such methods to provide a perspective for present efforts on classifying signals with complicated temporal and frequency characteristics, similar to the simulated datasets in Figure 1.2. Illustrations of the main approaches to signal analysis are shown in Figure 2.2, and each are discussed in turn.

Time-domain analysis (Figure 2.2, top left) uses the amplitude information of the signal in the analysis window. A typical time-domain method for anomaly detection and classification is that of matched filters [22, 23], but an underlying requirement is some degree of signal stationarity. Fourier-based methods (Figure 2.2,

top right) are ideally suited for the analysis of stationary signals whose durations exceed or are at least on the order of the analysis window length [1]. The drawback of the Fourier transform is the loss of temporal information, which is especially relevant for signals containing non-stationary or transitory characteristics (drifts, trends, abrupt changes, and beginnings and ends of events). The Fourier transform is also unable to “zoom-in” on temporal regions of interest in the measurements.

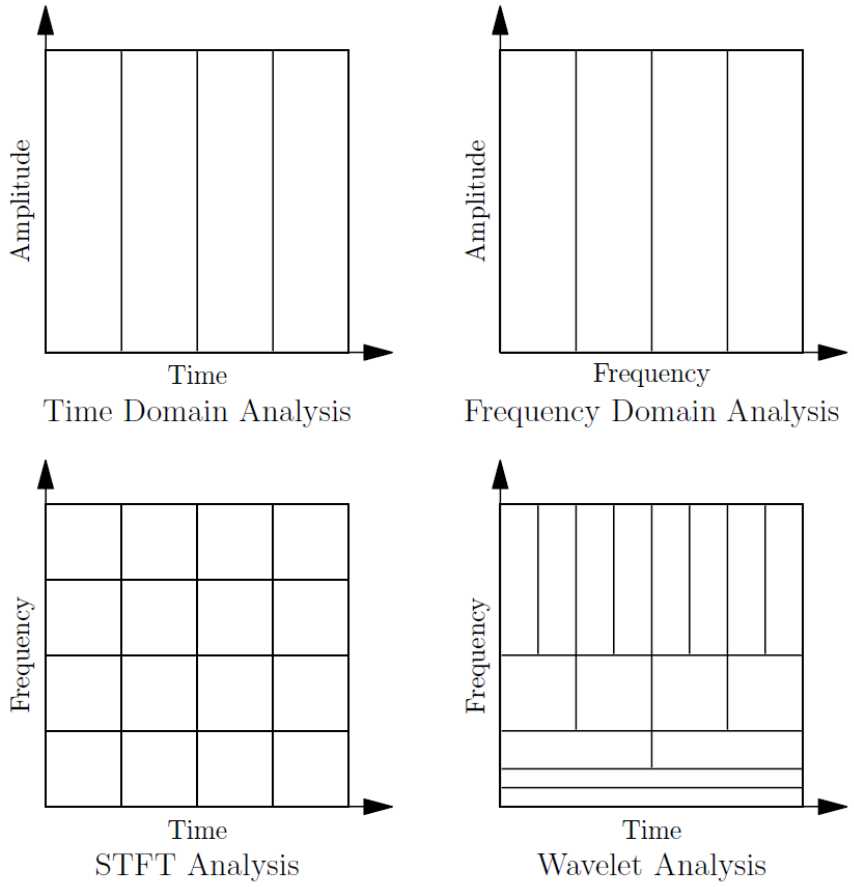


Figure 2.2: Pictorial of conventional signal processing approaches. Time domain processing (top left) uses signal amplitude information in the analysis window. Frequency domain processing (top right) uses Fourier transform information. The short-time Fourier transform (STFT) uses time-localized frequency information (bottom left) with fixed analysis window lengths. Wavelet methods (bottom right) employ variable analysis window lengths, resulting in adjustable scale (frequency) resolution.

In an effort to correct this deficiency, and motivated by the uncertainty principle in quantum mechanics, Gabor adopted the short-time Fourier transform (STFT) in 1946, analyzing only a windowed section of the signal at a time [24] (Figure 2.2, bottom left). The signal is mapped into a two-dimensional function of time and frequency, but time and frequency resolutions are limited and determined by the size of the window. Also, there is no flexibility in selecting different window sizes for variable frequency resolution.

Wavelet analysis generalizes the STFT by using a variable-sized windowing technique [25] (Figure 2.2, bottom right). The signal is mapped onto a time and frequency plane, where the frequency spread is inversely proportional to the time support. The completeness and orthogonally of a wavelet basis is represented by the non-overlapping tiles covering the time-frequency plane. The mathematical underpinnings of wavelet analysis date back to Joseph Fourier's theories of frequency analysis in the nineteenth century. The first recorded mention of a wavelet is found in an appendix to the thesis of Alfred Haar in 1909 [26]. The concept of wavelets in its present theoretical form was originally proposed by Jean Morlet and Alex Grossmann for application to seismic signal analysis [27]. Stephane Mallat was the first to develop an efficient way to implement the discrete wavelet transform algorithm using filters in 1988 [28]. Since then, research on wavelets has spread internationally, and is particularly active in the United States. Among scientists spearheading the effort are Ingrid Daubechies [29], Ronald Coifman [30], and Victor Wickerhauser [31]; Mallat presents an in-depth treatise on wavelet signal processing in his book [25].

Wavelet methods are by nature scalable, offer time-localized information, and are better for piecewise smooth functions [32]. They can reveal characteristics such as trends, discontinuities in higher derivatives, and self-similarity, and so appear to be most suitable to non-stationary RF analysis. Another advantage is the large number of wavelet families available, with varying individual strengths [25, 29]. Among many important wavelet properties, two of them will be used as descriptors later in the thesis and are briefly described below.

A first property is ***support width***. Most wavelets have **compact support**, which simply means the function is defined on a finite interval. Compact support guarantees the localization of the wavelets, which is important for processing distinct data regions. ***Energy efficiency*** describes how well a particular wavelet decomposition compacts a given signal. It specifically refers to the number of wavelet coefficients necessary to represent the signal, or a percentage of the signal. For example, a basis achieves 90% energy efficiency in 5 coefficients if a representation using the first 5 coefficients accounts for 90% of the signal energy. Additional theoretical details on wavelets are available in Mallat's treatise on wavelet-based signal processing [25], and will not be pursued in depth in this thesis.

2.2 Non-stationary (transient) wavelet analysis

Wavelet decomposition generates a different representation of one-dimensional data than traditional frequency domain techniques. This prompted its use in many applications, such as signal de-noising [33] or compression [25, 34, 35].

However, research on automatic detection and classification of transients has been limited. The framework of wavelet theory was established by the late 80s [27, 28, 36], and some of the early work dealt with problems where classes of transients were well characterized by prior parametric models [37]. The detection procedures resembled matched filtering in these cases. A second set of methods focused on problems where transient classes were not well characterized by prior models [3, 38], and this category is still of particular interest.

Underwater acoustic transient classification has been addressed by Desai and Shazeer [38], by using a three-level Daubechies wavelet packet transform to generate class-specific features for four training sets. Class-dependent frequency characteristics were ignored, and the choice of basis did not influence the feature selection process. In fact, much of the work in the area of transient classification is based on ad-hoc feature extraction, sometimes leading to more features than the original number of samples in the signal. The latter would be unfeasible for high data rates applications already suffering from very high dimensionality problems.

The first systematic wavelet feature extraction technique (for acoustic transients) was proposed by Rachel Learned [3, 39]. She reduced the dimensionality of the feature vector, while simultaneously accentuating the interclass distinctions. Specifically, Learned used singular value decomposition (SVD) on energy matrices for each class to detect the singular vectors carrying most of the information. The energy matrices were formed by concatenating energy vectors of wavelet decomposition coefficients for each given signal. The feature vectors for shrimp and

whale click data were tested in two classifiers, the nearest-neighbor and neural networks, and classification performance averaging 97% was obtained [39].

A disadvantage of wavelets basis methods is that the basis vectors may not optimally match a particular input signal structure. One alternative is custom designing a basis for the particular input. A specific technique for building a wavelet basis is Sweldens' lifting scheme [40]. The traditional method of defining a set of wavelets is to translate and dilate a particular function, the so-called mother wavelet. These are referred to as first generation wavelets. Sweldens uses multi-resolution analysis to design what have become known as second generation wavelets. His lifting scheme allows an infinite number of discrete bi-orthogonal wavelets to be generated from one that is fully customized and originally unique. A different architecture involving lifting factorizations with polyphase representations is shown by Brislawn in [41].

A second, more popular, alternative is using non-orthogonal, overcomplete (or redundant), wavelet sets, also called ***dictionaries***. Even though orthogonal transforms allow for fast signal representation via direct projection, and perfect recovery via direct inverse, dictionaries have the potential to lead to better energy efficiency (i.e., smaller number of necessary decomposition coefficients), and can be more adaptive to the data. Such dictionaries can be used adaptively either by selecting an orthogonal subset of the dictionary (i.e., basis) that is most suitable for the input data, or by obtaining sparse signal representations in the dictionary.

There are a number of published theoretical approaches to selecting a wavelet basis that is adapted to a given input signal. Most of these were developed for the purpose of optimizing signal approximation (or reconstruction), rather than classification [25]. One example is Coifman and Wickerhauser's best basis search method [2], developed mostly for efficient compression properties. It is an entropy-based algorithm that selects a best basis from a redundant set (i.e., a set with many more vectors than those needed to form an orthogonal basis), subject to minimizing the Shannon entropy as cost function. Other examples include "tree-based" methods, such as those described in [42, 43].

2.3 Transient classification

The study of pattern recognition has matured over the years, and comprehensive studies of general classifiers (again, algorithms which evaluate features and assign a signal class), including their strengths and underlying problems, have been published [21, 44]. Depending on the specific application, there are various pattern recognition methods available. Bayesian decision theory is a fundamental statistical approach and its origins date back to the mid 1700s [45]. In the 1980s, there was a dramatic growth in research and applications of Bayesian methods, and currently they are widely accepted and used [46]. However, they rely on knowledge of the underlying probability densities of the signal classes, which is rare in practice. When little is known about the probability structure, a range of classification

techniques emerge. Disciplines such as machine learning, statistics, and neural networks continually expand the foundations of pattern recognition.

Methods used in conjunction with classification of transients in recent work encompass the older, more established classifier techniques [4, 37-39, 47-50]. Some examples are Maximum-Likelihood estimation, nearest-neighbor rule, and linear discriminant functions. Neural Networks (NN) can be suitable for classification problems with significantly overlapping patterns, high background noise and dynamically changing environments. Probabilistic neural network (PNN) is also a popular technique, and has the advantage of fast training and inherently parallel structure [21]. However, PNNs can only yield optimal performance if a sufficiently large training set is provided. Variations of the NN or PNN techniques are extensively used in the more recent publications on transient classification [3, 4, 38, 47, 50]. There are classification software packages available online, implementing established methods, such as the Classification Toolbox [51] accompanying [21], LIBSVM [52], and Weka [53], and the latter will be briefly used in Chapter 5.

Several wavelet-based classifiers implemented for real applications of transient analysis have demonstrated various success rates, e.g., [4, 47, 48]. Caffrey explored the classification of impulsive RF events in the earth's atmosphere [4]. An optimal mother wavelet was computed adaptively via a neural network (NN). The feature vectors obtained by decomposition were used as input to a two layer feed forward NN. Although the training set was classified without error, the algorithm performed extremely poor compared to a FFT-based classifier in a real test scenario.

A similar approach was presented by Angrisani [47] for voltage spike classification. He chose a modified Morlet wavelet transform and fed the output coefficients to a two layer NN. The method successfully enhanced the classification performance by using a smaller feature set. Crouse developed a framework for statistical signal processing based on wavelet-domain Hidden Markov Models (HMMs) [48]. An expectation-maximization algorithm fitted the HMMs to data, but no extension to an actual application was made. A more practical approach was proposed by Shin [54] to detect a transient signal component of interest from a composite signal waveform, by extending conventional time-domain matched filtering to time-frequency domain optimal filtering.

This thesis specifically addresses the classification of non-stationary, pulsed, RF signals. This particular application has not been extensively researched, and only in the past few years has it become of interest. C.H. Lee published a review of wavelet use in power engineering applications [49]. Most of the work is very basic and applies to characterizing power quality disturbances in distribution systems using low-level Daubechies wavelets. Perera [50] shows a classification system for fault vs. non-fault transients in three-phase power distribution systems using a probabilistic neural network technique, on data generated via electromagnetic transient simulation software, but it is tailored for a specific system. Some classification work was done on the Fast On-orbit Recording of Transient Events (FORTE) satellite data recorded by Los Alamos National Laboratory [55, 56], which in many ways is very similar to the simulated RF data considered in this thesis. The FORTE work in [56] employed a

genetic algorithm named Zeus for extracting features, which were used in a support vector machine (SVM) classifier. Although classification accuracy was 100% on training data, performance was unsatisfactory on test data.

2.4 Dictionary methods

Considering the illustrations in Figure 2.2, the respective signal space (e.g., frequency vs. time in the STFT case) is completely covered with non-overlapping processing tiles (e.g., 3x3 of them in the STFT case). When orthogonality is no longer enforced, tiles will cover the signal space in an overlapping and/or incomplete fashion, resulting in what is called a ***dictionary*** instead of a basis. Almost any matrix could be called a dictionary. A somewhat similar notion to that of a dictionary is the notion of frame [5, 57], which in finite dimensions is just a dictionary having linearly independent rows. Frame constraints will not be specifically enforced in this thesis.

Visually, dictionaries that cover the signal space completely with overlapping tiles are called overcomplete dictionaries. Dictionaries that do not cover the entire space, but are concentrated in the areas where the input of interest lies, with or without overlapping tiles, are undercomplete dictionaries. Another way of describing completeness of a dictionary is by comparing the number of vectors (i.e., elements) in the dictionary to the natural dimensionality of the input vectors for a particular application. A larger number of elements than the natural dimensionality means the dictionary is overcomplete, and a lower number of elements implies an undercomplete dictionary. The degree of dictionary completeness (i.e., dictionary

size) has not been explored in depth prior to this thesis and was identified as a fundamental issue to be studied and solved in the closing remarks of [58]. While *overcomplete* dictionaries have been the common choice in most of the recent image processing publications, this thesis is focused on evaluating *undercomplete* learned dictionaries for classification, which is a novel use of such dictionary methods.

2.4.1 Dictionary search algorithms

Since a dictionary is not a basis, decomposition of an input is no longer possible via straightforward projection. In fact, a decomposition is no longer unique, and it typically does not imply perfect reconstruction. Much of the recent focus has been on algorithms to compute adaptive signal representations on these redundant dictionaries. These algorithms, usually called ***pursuit algorithms***, search for efficient, but non-optimal representations (i.e., signal approximations). Mathematically, the goal is find a good enough sparse approximation, *a*, of some given input, *x*, using the dictionary *D*, which can be written as:

$$\min_a \|x - Da\|_2^2 + \lambda S(a), \tag{2.1}$$

where the l_2 norm is defined by

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{1/p}, \text{ for } p > 0. \tag{2.2}$$

The first term of equation (2.1) controls the mean square reconstruction error, and the second is used to enforce some constraint on *a* (i.e., it is some norm of *a*). In modern day use of dictionaries, the underlying requirement for the approximation

vector, *a*, is ***sparsity***. A sparse vector is defined to be one with few significant coefficients; such vector is called *L*-sparse if it has at most *L* non-zero entries. The quantitative definition of sparsity (i.e., what percent of the total number of coefficients should be zero for the vector to be even referred to as sparse) varies; in this thesis vectors are considered to be sparse if at most 25% of their entries are non-zero. The l_0 "norm" was first used by Donoho [33] as a direct measure of sparsity, since it counts the non-zero vector entries; it is not a true norm in the mathematical sense, hence the quotation marks. Given that the dictionary *D* is overcomplete, the decomposition *a* is not unique, and several methods for finding a good solution for *a* have been proposed.

One of the first approaches to solving equation (2.1) was the Method of Frames (MOF) [59], which uses an Euclidian norm on *a*. That is, the approximation solution *a* is the one whose coefficients have minimum l_2 (Euclidian) norm. It is also sometimes called a minimum-length solution. One main problem with MOF is that it does not preserve sparsity, i.e., if the underlying input has a sparse representation in terms of actual dictionary elements, the vector of coefficients found by MOF is likely very much less sparse.

A second approach was presented by Mallat and Zhang [6], directly addressing the issue of sparsity. The sparsity-inducing norm *S(a)* in equation (2.1) is considered to be the l_0 "norm," which is not differentiable and therefore standard minimization techniques cannot be employed. Their method, called matching pursuit, is a greedy-type algorithm, which means that it chooses at each iteration a waveform

to best approximate the current signal residual (i.e., local optimization). The approximations obtained with matching pursuit can be improved using Pati's orthogonalization refinement [60], however, the added computation cost can be high.

The specific term 'sparsity' was first used by Chen, Donoho, and Saunders in a 1995 Stanford report [61]. In this work, they made a key contribution to the field by demonstrating a convex optimization using an l_1 norm in equation (2.1) that led to a sparse solution a. Chen and Donoho further developed this theoretical method of obtaining signal representations in overcomplete dictionaries using global convex optimization, known as atomic decomposition by basis pursuit, and provided several updates, with the latest one in 2001 [62]. This algorithm guarantees a sparse approximation, yet its computational cost can be high. However, Chen and Donoho's analysis provided the theoretical backbone for much of the research on sparse approximations that followed.

The fast growing field of compressive sensing has its roots in sparse approximation research. Although an early paper having compressive sensing undertones was published in 1981 by Levy [63], the basic compressive sensing theory and terminology emerged in 2006 in the works of Candès [64, 65] and Donoho [66], and provided a turning point for research on sparse approximations. Compressive sensing uses the sparsity of natural signals to sidestep the Nyquist sampling rate bounds, and still obtain perfect reconstruction from a *limited number of linear measurements* (i.e., compressive sampling). Since then, research on the subject has expanded, resulting in an extensive number of publications from many academic

groups. Of interest for this thesis is Chartrand's method [67, 68], of nonconvex optimization using a regularized l_p norm with $p<1$. He demonstrated that an l_p norm can lead to even sparser solutions than l_1, and has lower reconstruction error [69]. This l_p norm is an avenue for future work and will be further discussed in Chapter 7. Compressive sensing is not the focus of this thesis, but it is mentioned to provide the reader with a deeper appreciation of the impact sparse representations has had on the signal processing community. It also helped inspire one of the questions posed in this thesis: specifically, do the learned dictionaries really need to be overcomplete for signal processing tasks such as classification, or can they be undercomplete?

Modern signal processing techniques frequently rely on signal representations that are adapted to the data. For many applications, e.g., compression and denoising, representation sparsity is also desirable [70]. Such sparseness can be achieved either by thresholding representation coefficients [33], or by forming approximations in overcomplete dictionaries via methods such as l_1 "lasso" [71] or l_0 matching pursuits [6, 60].

Sparse representations over redundant dictionaries have led to state-of-the-art results in audio processing [7], fundamental image processing tasks, such as denoising [72, 73], restoration [74-76], compression [77], reconstruction [78, 79], classification [80, 81], as well as video restoration [76], and analysis of hyperspectral imagery [82]. These sparse approximations are based on the idea that natural data is highly structured, and can therefore be compactly expressed using sparse linear combinations of dictionary elements. An in-depth look was taken at obtaining the

sparse coefficient vector, *a*, in equation (2.1), but not much has yet been said about the dictionary, *D*, which is a key consideration. Dictionary design efforts in published literature are centered on two dictionary types: parametric and learned.

2.4.2 Parametric dictionaries

In the case of parametric dictionaries, a mathematical model (i.e., generating analytical function) must be formulated first. This leads to dictionaries that are highly structured, with degrees of freedom controlled by the generating function. Such dictionaries are described implicitly by their underlying model, rather than explicitly by the actual matrix of dictionary elements, and can be generated (and re-generated) fast using a numerical implementation. There are two ways to create parametric overcomplete (non-orthogonal) dictionaries. The first one is to oversample the parameter space of the generating function. The second one is by merging dictionaries to make bigger, more "expressive" dictionaries (e.g., Heaviside with Fourier [62]). Designing parametric dictionaries is an undergoing research effort, and dictionary families of interest include chirped Gabors [6, 7], Wavelets [5, 25, 83], Curvelets [84, 85], Shearlets [86, 87], Contourlets [88], Ridgelets [89], and Bandelets [90], among others. For the specific RF data of interest in this dissertation, the chirped Gabor dictionary presents the closest match to the signal properties; its use for adaptive feature extraction will be explored in depth in Chapter 3.

2.4.3 Learned dictionaries

Parametric dictionaries with analytical, closed-form elements impose assumptions about the underlying structure of the data. On the other hand, learned or adaptive dictionaries avoid this constraint, and are explicitly defined by the matrix of dictionary elements. Initially introduced by Olshausen and Field [9, 91, 92] for modeling the mammalian visual cortex, the idea of learning a dictionary directly from data has gained momentum in the image processing field. State-of-the art dictionary learning techniques are inspired either by neuroscience [93, 94] or by codebook design (vector quantization) approaches [10, 11]. Such learned dictionaries have been used for many image-based applications, e.g., texture classification [15], multispectral satellite imagery analysis [13], image compression [77, 95], and to model the visual cortex [96], among others. Examples of dictionary learning techniques include Aharon's K-SVD [10], Wright's SRC [97, 98], Mairal's online learning [99], Jenatton's tree embedded dictionary [100], and Skretting's RLS-DLA [95].

All the learned dictionary work summarized above has some mathematical formulation for learning dictionary elements directly from the data input. Different algorithms, for extracting data-specific features for object recognition in images, are based on the image statistics and geometric correspondences, such as the Scale Invariant Feature Transform (SIFT) [101], finding the "gist" of a scene [102], spatial pyramid match [103, 104], among others (e.g., [105-107]).

Sparsity-inducing learned dictionary techniques have led to significant improvements in image representation, classification, and restoration (e.g., [14, 100,

108-110]), while also advancing understanding on sparse approximations [80, 111]. Chapters 4 and 5 of this dissertation will focus on extending and adapting learned dictionary methods to RF detection/classification, and will introduce novel approaches along the way. Chapter 7 will demonstrate a novel application of undercomplete learned dictionaries to multispectral satellite imagery classification.

3. Sparse Representations in Overcomplete Parametric Dictionaries

One approach to RF target detection and classification involves using parametric (or analytical) dictionaries that are overcomplete with respect to the natural input dimensionality of the data (i.e., the length of the analysis window). Using pursuit-type searches in overcomplete dictionaries leads to sparse approximations, and this approach can work well for classification of target signals that share the same function class (e.g., chirping signals) as the dictionary elements. The precursor of the work in this chapter is found in [8], where a different simulated data set was used in a hierarchical classification system. Features obtained from a parametric overcomplete dictionary were compared to STFT-based ones in a classification setting using various Weka classifiers [53]. The results showed that features extracted from the overcomplete dictionary had greater classification potential, and motivated this subsequent study of parametric dictionaries in an effort to improve the quality of the extracted features. For this purpose, Chapter 3 focuses on attempting to segment the target signal from the other components, and analyzes the effects on features observed during data windowing, changes in noise levels, and increase in amount of clutter, among other things.

3.1 Feature extraction algorithm

3.1.1 Dictionary matching pursuit search

In the case of orthogonal bases (e.g., Fourier Transform), a discrete input signal of natural dimensionality *N*, $x \in \mathbb{C}^N$, is simply projected to obtain a vector of representation coefficients. An overcomplete dictionary, $D = [d_1 | d_2 \ldots | d_K] \in \mathbb{C}^{NxK}$, either learned or parametric, must instead be searched iteratively in order to find a coefficient vector, *a*, such that the approximation *x=Da* meets a specified criterion. For sparse approximation, the solution vector *a* will have a large concentration of energy in few of its coefficients, and will give small errors relative to noise in the signal *x*. The problem of sparse approximation can be approached in two ways: either minimizing the number of coefficients in *a* such that the reconstruction has a maximum error $\varepsilon > 0$ (equation 3.1), or alternatively as minimizing the error of the model using a maximum number of atoms *m>0* (equation 3.2). In general, these two approaches will lead to different solutions for the sparse vector *a*.

$$\min \|a\|_0 \text{ such that } \|x - Da\|_2^2 \leq \varepsilon, \tag{3.1}$$

or

$$\min \|x - Da\|_2^2 \text{ such that } \|a\|_0 \leq m. \tag{3.2}$$

Here, the "norm" $\|a\|_0$ is, as mentioned in Chapter 2, a pseudo norm and represents the number of non-zero elements in *a*. To ensure there exist solutions to *x = Da*, one typically uses a dictionary of size *K >>N* with *rank(D) = N*, which means that the dictionary is overcomplete for the inner product space $\mathbb{C}^N$. The solution, *a*, is

therefore not unique, and the one most favorable with respect to equations (3.1), (3.2), or with respect to the signal "descriptiveness" [112] must be selected.

Finding *a* from equations (3.1-3.2) is an NP-hard (i.e., non-deterministic polynomial-time hard) problem, and many methods have been proposed and extensively studied to find an approximate solution, as briefly mentioned in the previous chapter. One approach is to replace the l_0 "norm" in (3.1) or (3.2) with another one, e.g., an l_1 norm [62, 113], an l_2 norm [59] or an l_p norm [67]. These approaches, however, are costly for high-dimensional signals and the large associated dictionaries. Greedy iterative descent methods (i.e., methods that iteratively select a locally optimal choice with the hope of finding a global optimum), provide another approach to solving the sparse problem above for high-dimensional signals and large dictionaries. Matching pursuit (MP) [6] is the simplest of these, and can be efficiently implemented for large shift invariant dictionaries and high-dimensional data, e.g., [7, 114]. This is the primary dictionary search method used in this thesis, and will be explained in greater detail in Section 3.1.3. Other iterative approaches provide lower approximation error, e.g., orthogonal MP (OMP) [60, 115, 116], orthogonal least squares (OLS) [117], cyclic matching pursuit (CMP) [112, 118], but usually incur higher computational costs than MP.

3.1.2 Multi-scale chirped Gabor dictionary

Research in the field of adaptive audio signal processing [7, 119] has led to methods which can represent non-stationary audio signals using a Gaussian multi-

scale chirped Gabor dictionary. This dictionary *D* consists of the set of complex atoms *g* given by:

$$D = \left\{ g_{u,s,f,c}(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) \exp\left(2\pi i\left(f(t-u) + \frac{c}{2}(t-u)^2\right)\right)\right\}, \tag{3.3}$$

where the window function, *g(t)*, is unit Gaussian, shifted by *u* and scaled by *s*. The parameter *f* is the frequency in natural units with scale dependent frequency resolution, and the *c* parameter controls the linear chirp rate of the atom. Theoretically, such atoms can be used to approximate both continuous wave signals (atoms with *c*=0), as well as other types of chirped signals, (e.g., with exponential or logarithmic chirp), by using piece-wise fitting with linear chirp atoms. Example chirped Gabor dictionary elements with *N*=512 samples and varying compact support are shown in Figure 3.1. This dictionary is selected to provide a feature extraction comparison for the RF simulated data introduced in Section 1.1 using parametric overcomplete dictionaries.

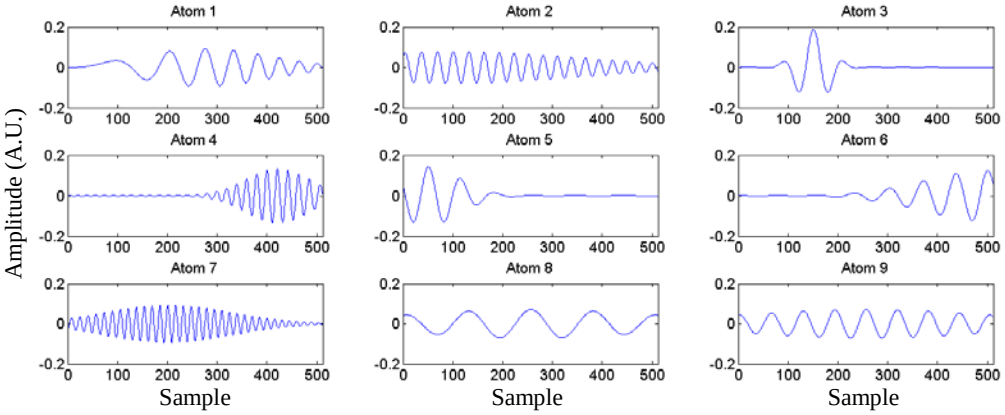


Figure 3.1: Example timeseries of chirped Gabor atoms. Plots show the real magnitude of few dictionary atoms as a function of sampled time. While all atoms are timeseries with *N*=512 samples each, the support length for each atom is different (e.g., atom 4 has support length of only ~250 samples, atom 3 has support of ~150 samples, atom 2 has support on all 512 samples).

a) Dictionary size

Extending the use of this multi-scale Gabor dictionary to RF applications poses computational problems, due to the much higher sampling rates compared to audio applications, and by extension the much higher order dimensionality. The necessary dictionary size for overcompleteness, *K*, is large, and a direct search of the 4 dimensional parameter space (*f, u, s, c*) would not yield results in real-time. For example, given a sampling rate of 1MHz, and an analysis window size of 512 samples (0.5 ms), an example overcomplete dictionary of non-chirping atoms with 512-sample length support would have a minimum size of 3350 atoms, and a possible maximum size that would depend on the user. This count does not include the possible chirping atoms, since the range of permissible chirp rates for a specific atom depends upon its exact base frequency *f*, scale *s* (i.e., over how many samples of *t* is the frequency changing), subject to the Nyquist theorem. Given an atom (f_0, u_0, s_0, c), its instantaneous frequency is obtained from equation (3.3) as the derivative of the complex phase:

$$\omega(t) = 2\pi(f_0 + c(t - u_0)), \tag{3.4}$$

where $\omega \in (0, 2\pi]$. The range of possible values for the chirp rate, *c*, i.e., chirp rates not inducing aliasing given the bandwidth *B*, is limited to $\left[-\frac{f_0}{s/2}, \frac{B - f_0}{s/2} \right]$ for positive chirp rates, and $\left[\frac{f_0 - B}{s/2}, \frac{f_0}{s/2} \right]$ for negative chirp rates. That is, the range of

possible chirp rates for a given atom frequency and scale is bound by a rhomboid in the chirp rate vs. frequency phase space plane.

b) Reduced dimensionality search

The dictionary, *D*, can be searched with reduced computational cost by using Gribonval's fast *O(NM)* ridge-pursuit algorithm [7], where *M* is the number of atoms desired in the decomposition. The ridge-pursuit algorithm is a two-step pursuit, and explores the local maxima property of the discrete, non-chirping, Gabor dictionary subset $D^G = \{g_{u,s,f,0}(t)\}$. One of the main findings in [7] is that the ridges of local maxima of D^G correspond to the ridges of maxima in D. This allows one to first select the best matching non-chirping, complex, Gabor atom $g(s_m, u_m, f_m)$ from D^G. Then, the corresponding atom neighborhood in *D* is searched to select a locally optimal chirped atom, $g(s_m, u_m, f_m, c_m)$. For real input signals, the optimal complex atom is used to find the equivalent real-valued chirped Gabor atom using 'dual molecules.' These dual molecules (or dual bases) are computed from the complex chirped atom and its conjugate atom using the formulation in [7]. Much insight into the workings of Gribonval's algorithm was obtained from the expanded version in [120]. The fast ridge pursuit search over the chirped Gabor dictionary can falsely introduce conjugate chirp rates as atoms are sequentially subtracted from the signal. To mitigate this effect, in this dissertation an additional step was added once the locally optimal real-valued chirped atom was found. Specifically, the non-chirped atom and the chirped atom were compared one last time in terms of their respective inner products with the

input vector. The atom yielding the highest inner product was selected as the final dictionary atom at the respective matching pursuit iteration.

3.1.3 Target classification algorithm

Given the dictionary *D* with *K* atoms, and a data window of length *N* samples, where $N<<K$, *M* dictionary atoms are iteratively selected that minimize the signal residual at each step using a simple greedy MP algorithm, briefly detailed here. At the first iteration, the real atom giving the largest inner product with the signal is found using the fast-ridge pursuit in 3.1.2. The contribution of this atom is then subtracted from the signal, and the process is repeated on the residual. This continues until some predetermined stopping point (e.g., number of atoms or size of the residual), as shown in equation (3.5) below.

$$\begin{aligned} R_0 &= x \\ d_{k,m} &= \max_D \left(\left| \left\langle R_m, d_k \right\rangle \right| \right) \\ R_m &= \left\langle R_m, d_{k,m} \right\rangle d_{k,m} + R_{m+1} \end{aligned} \tag{3.5}$$

Thus, unlike an orthogonal basis in which all feature vectors represent the same basis elements, identically ordered for every time window, the atoms selected by greedy pursuit can differ from time window to time window, and the exact atom ordering can also be different. This means a classification scheme based on atomic ordering would not be a correct choice. The *parameters* of the matched atoms are therefore selected as classification features, specifically the frequency, *f*, and the chirp rate, *c*. In anomaly detection applications, the time centers, *u*, of matched atoms could also be used to pinpoint the exact location of the anomaly. The features obtained

using matching pursuit over the dictionary, *D*, can be used to train standard classifiers in the Weka collection [53] as previously seen in [8].

3.1.4 Match confidences

Section 3.2 evaluates the effects of windowing, noise, overcompleteness, and clutter on the extracted atoms. These effects are quantitatively assessed by binning (counting) the atoms returned by MP in confidence regions around the true signal components. For example, given the target signal component, concentric confidence regions around the target location in the (*frequency*, *chirp rate*) plane are considered, and the number of matched atoms in each region is counted. The total number of matched atoms for a given region is then compared to the total number of possible true target observations (i.e., the number of ON analysis windows) and used to generate a *match confidence*. Similarly, match confidences for all the other signal components are also calculated. Confidence regions of up to 25 kHz in diameter are considered to generate frequency match confidences, and up to 15MHz/s for chirp rate confidence. Given that most of the base frequencies present in the data are separated by ~60 kHz, a maximum confidence region of 25 kHz around each would diminish the possibility that the counted atoms belong to frequencies different from the frequency of interest. The resulting confidences are evaluated in two ways.

First, they are aggregated into curves as a function of the size of the confidence region for each signal component, disregarding the order in which the atoms were returned by the MP algorithm. Such curves are reminiscent of receiver-

operating-characteristic (ROC) curves, in that they show how many correct parameter matches (i.e., "true positives") were found as the confidence region grows incrementally. These ROC-like curves give a visual assessment of the ***fidelity of the returned atoms*** (e.g., how close they are to the true frequency).

Secondly, the maximum match confidence is calculated as a function of the atom order of return for each signal component. That is, given only the atoms of order *n* (i.e., the *n*-th returned atoms for every analysis window), the maximum match confidence for every signal component is determined using the largest confidence region. The confidence as a function of atom order gives a measure of how good the atom match is in terms of the amount of signal component it captures.

3.2 Effects of windowing, noise, overcompleteness, and clutter

In order to better ascertain which effects dominate the quality of extracted atoms, initial studies are performed on reduced complexity simulation test data, containing:

- Target signal at a 220 kHz base frequency with chirp rate 15.5 kHz/s and relative amplitude 3.
- Four CW clutter emitters at 60 kHz, 120 kHz, 310 kHz and 430 kHz, with signal-to-clutter amplitude ratios {6, 5, 3, 4} (i.e., the CW clutter is not overlapping the target spectrum of 220 kHz – 297.5 kHz).
- Additive white noise, with signal-to-noise amplitude ratio equal to 1.

The Far CP data case will be considered after the reduced complexity case to ascertain the changes introduced by having an additional chirping source and more CW clutter. In terms of amplitudes of the CW clutter signals, they are chosen to be nontrivial even in the reduced complexity case, i.e., higher than the target amplitude.

These effects are assessed strictly in terms of the fidelity of the returned atoms (i.e., how well the parameters of the atoms match the true signal components), and their viability for use as features in a target detection and classification setting. Specific ways of assessing fidelity will be detailed in Section 3.2.1. Reconstruction of the input timeseries using the chirped Gabor dictionary with the fast ridge pursuit is not a metric pursued in this work, as it is explored in detail in [120]. A brief discussion of reconstruction with this dictionary will be given instead in Section 3.4.

3.2.1 Data windowing

Methods of matching pursuit can introduce artifacts at the edges of the analysis window as atoms are sequentially subtracted from the signal. For the RF data explored, these edge effects manifest in atoms of similar frequency and chirp rate being extracted repeatedly from the analysis data at different locations. The specific MP extraction pattern observed consists of the algorithm matching an atom in the center of the analysis window, followed by a left side atom match with comparable frequency and chirp rate, and then similarly followed by a right side match.

Classical signal processing relies on windowing techniques prior to the MP search to help mitigate such edge effects. Three standard data windowing functions,

Gaussian, Hamming, and Hanning are compared based on the quality of the atoms returned by the MP for a selected timeseries. In order to distinguish between effects of windowing and possible effects of noise and clutter, the test cases considered are first the reduced complexity case with SNR 3:1, followed by the Far CP with SNR 3:1 data set. That is, changes in SNR and the position of the chirped clutter are not taken into account at this time. The number of extracted atoms is set to 25 for every analysis window.

Figure 3.2 shows a scatter plot of chirp rate vs. frequency for the first 25 extracted atoms, colored according to their order of return by the pursuit search and aggregated over ON (left panel) and OFF windows (right panel) of reduced complexity data. This type of plot is useful because it quickly shows if the returned frequency and chirp rate parameters are close to true signal components. The circles in Figure 3.2 represent the true signal components, and their diameters are proportional to the respective amplitudes. As expected, the order in which the atoms are returned (red for first 1-5 atoms, purple for last 20-25), which is a direct result of the amplitude of the matched atom, closely matches the sequence of relative amplitudes for the signal components. Atoms corresponding to the CW components are grouped in bands parallel to the *y*-axis (e.g., group enclosed by rectangle in Figure 3.2), with abscissas approximately equal to the CW frequencies. That is, the frequency parameter of the atoms appears to be a good feature for discriminating CW data. The chirp rate estimation presents significant smearing and appears highly unreliable for the CW components, as it introduces false positive and negative chirp

rates with variations spanning several decades. Note that some of the first 5 returned atoms (shown in red) correctly have zero chirp rate and are positioned within the circles marking the respective signal components.

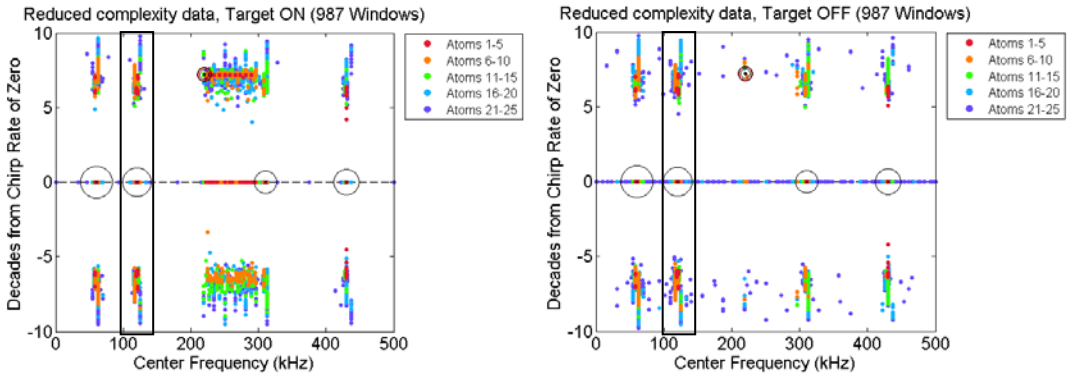


Figure 3.2: Scatter plots (chirp rate vs. frequency) for reduced complexity data. Shown are the first 25 extracted atoms, aggregated over ON windows (left) and OFF windows (right), and colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane. Rectangle highlights group of atoms with frequency around 120 kHz.

For the chirped target pulse, the returned atoms cover the frequency spectrum of the target, between 220 kHz and 297.5 kHz, and are aligned along both the true chirp rate, as well as the conjugate chirp rate and the zero-chirp axis. The lack of an exact frequency match along the true chirp rate area is to be expected, as the length of an analysis window does not cover an entire target pulse, but rather portions of it. The instantaneous target frequency estimated by the atoms is therefore increasing across consecutive analysis windows proportional to the chirp rate.

Figure 3.2 shows that the fast ridge pursuit over the Gabor dictionary might be able to generally match the frequencies in a given analysis window, and could be

suitable to piece-wise approximate (i.e., reconstruct) the input signal, as claimed in [7]. However, using the features of the returned atoms for a classification problem does not appear to be a straightforward procedure, due to the erroneous chirp rate estimation.

When windowing is applied to the reduced complexity case, there is no apparent improvement in the frequency match confidence as a function of atom order. This is seen in Figure 3.3, which shows how close the frequencies of the returned ***ordered*** atoms matched the frequencies of signal components in the data set. The CW components are arranged in the legend in decreasing order of their amplitudes. For every signal component, the confidences exhibit repeated peaks, i.e., the subtracted atoms do not fully capture a signal component in one match, especially in the windowed cases, and multiple MP iterations are needed. The target component has the worst frequency match compared to the CW components, and the highest confidence rate is actually achieved in the non-windowed case, followed by the Hamming window case. Given that the Gabor atoms do in fact have a Gaussian envelope, it was expected that the Gaussian window would at least outperform the non-windowed case, if not all the other windowing functions. The performance of the Hamming window can be explained by considering Figure 3.4.

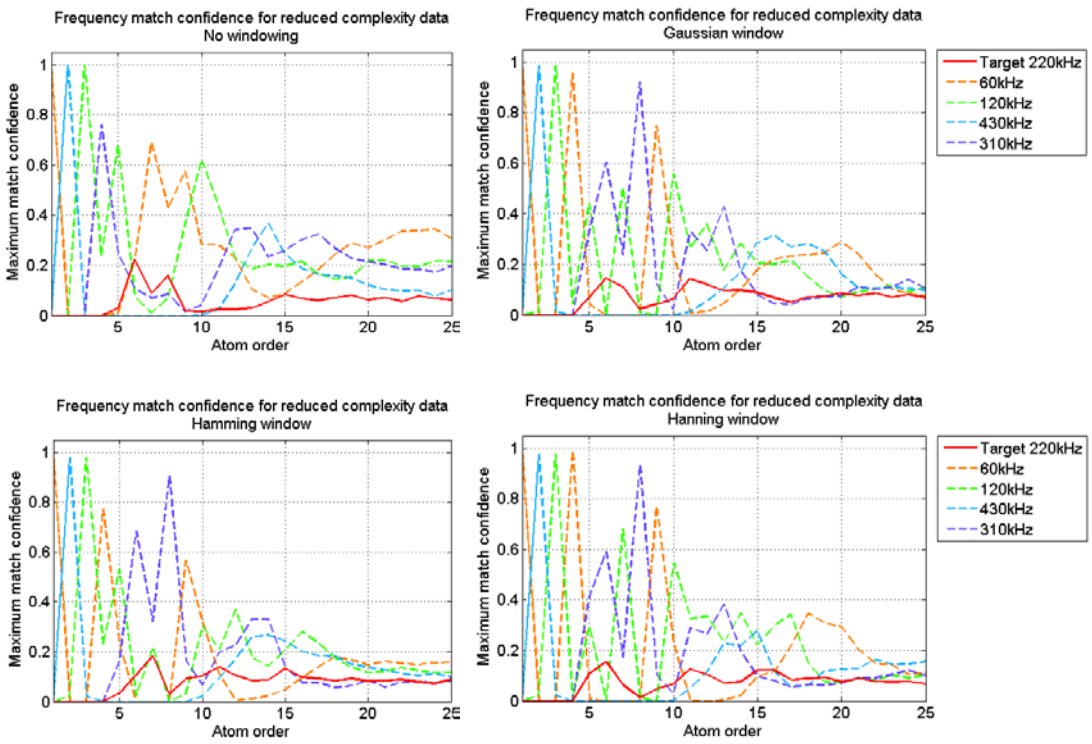


Figure 3.3: Match confidences for reduced complexity data (parameters given in the text) with various pre-windowing functions. The CW components are arranged in the legend in decreasing order of their amplitudes. Windowing does not appear to improve the frequency match confidence as a function of atom order.

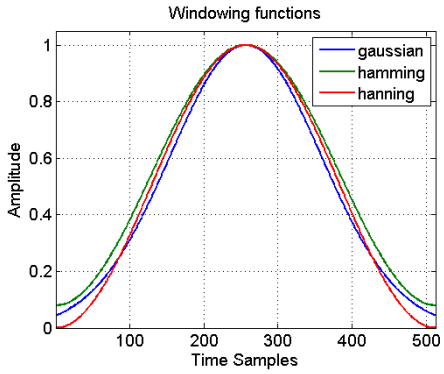


Figure 3.4: Windowing functions used on the timeseries. The Hamming window has the highest amplitude in the tail sections, i.e., retains the most signal at the edges of the analysis window, while the Hanning window has the most attenuation in the tail sections.

Here the amplitudes of the three windowing functions are plotted, and the Hamming window (green trace) has the widest of the bell-shaped curves, and the largest tail-section amplitude. That is, the Hamming window retains more of the signal's time data, which could perhaps explain the slightly higher quality of the extracted atoms.

For the chirp rate confidence match, Figure 3.5 shows that windowing only marginally improves the quality of the chirp rate estimate in the case of the first batch of target returned atoms, after which the confidence in the windowed case degrades significantly compared to the non-windowed case. The sign of the estimated chirp rate in Figure 3.3 is taken into account, i.e., only positive chirp rates are considered, matching the target chirp rate sign.

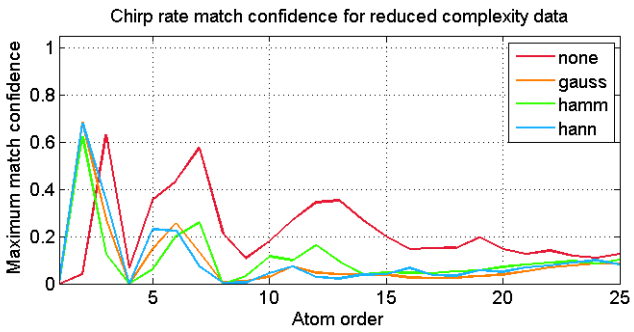


Figure 3.5: Chirp rate confidence for reduced complexity data, with and without windowing.

The three windowing functions are now applied to analysis windows of the more complex data set Far CP with SNR 3:1 and the resulting first 25 atoms are used to generate match confidences. Recall that windowing is performed in order to evaluate whether it would improve the quality of the returned atoms (i.e., the fidelity of their parameters). Figure 3.6 again shows the scatter plot of chirp rate vs.

frequency for the first 25 extracted atoms, colored according to their order of return and aggregated over ON (left panel) and OFF (right panel) windows. As the competing chirped pulse emitter and the additional CW clutter are introduced back in the data, the frequency match seems to improve somewhat (i.e., there is more alignment along the true frequencies compared to Figure 3.2). The chirp rate estimation continues to be problematic, however, and the false conjugate chirp rates appear for every signal component.

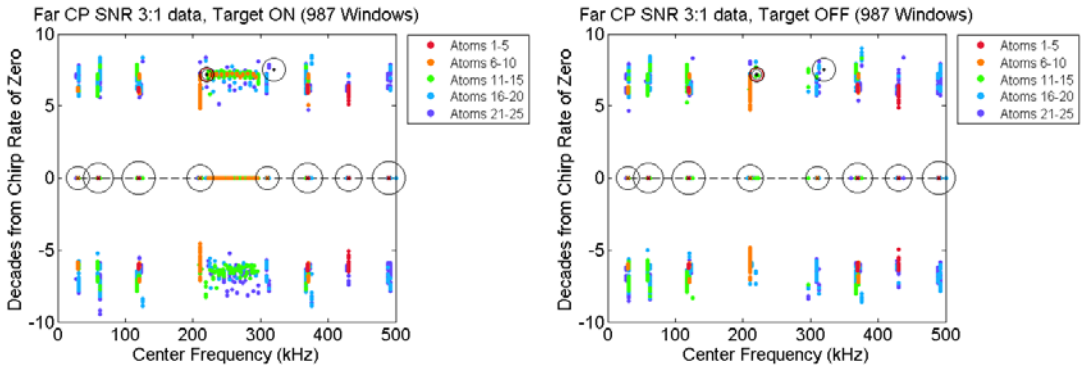


Figure 3.6: Scatter plots (chirp rate vs. frequency) for Far CP data. Shown are the first 25 extracted atoms, aggregated over ON windows (left) and OFF windows (right), and colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane.

The four panels of Figure 3.7 show the frequency match confidence as a function of atom order for the Far CP data in the non-windowed case (top left), as well as the Gaussian window case (top right), Hamming case (bottom left), and Hanning case (bottom right). This comparison is useful because it allows us to immediately see that the highest match confidence among the windowed case is observed for the Hamming window, as previously noted for the reduced complexity data.

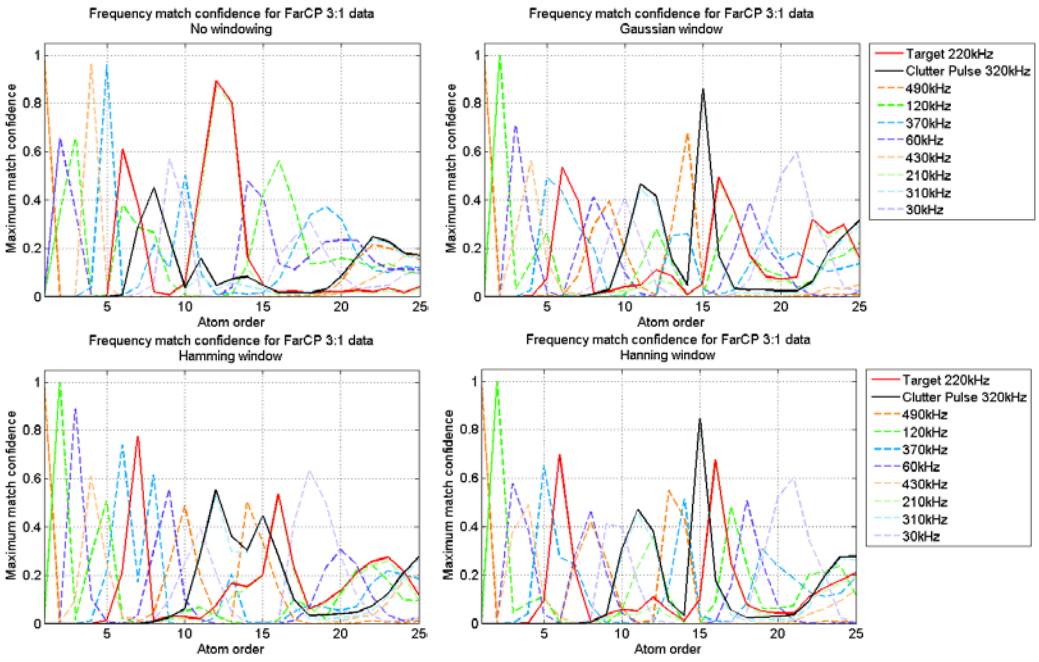


Figure 3.7: Match confidences for Far CP data with various pre-windowing functions. The CW components are arranged in the legend in decreasing order of their amplitudes.

Windowing the data in this more realistic case impacts first of all the *order* in which the first atoms for each component are returned. Secondly, the match confidences for the chirping signal components (i.e., the target pulse and the clutter pulse) are higher in the Hamming and Hanning case for the first returned respective atoms. Across all atoms, there is a better frequency match for the chirped clutter pulse in the windowed cases compared to the non-windowed case, but the best target match is still achieved in the non-windowed case. Overall, 5 signal components are matched with better than 0.8 confidence in the non-windowed case, compared to 4 in the Gaussian case, 3 in the Hamming case, and 4 in the Hanning case. That is, even though Hamming windowing achieves the highest target confidence among the windowed cases, it does not appear to perform as good for the other signals as the

Gaussian window. Compared to the non-windowed case, the windowed cases seem to perform more poorly in terms of atom features.

To further strengthen this observation, the fidelity of the returned atoms in the Far CP case is now explored in Figure 3.8 for the non-windowed case and the three windowed cases. The atoms corresponding to the CW frequencies have good frequency fidelity, i.e., less than 2 kHz deviation from the true frequency, in all four cases. In the chirped component case, the fidelity is not nearly as good, and remains greater than 7 kHz in all cases. Figure 3.8 shows again that, while windowing increases the chirped clutter confidence, it decreases the target confidence.

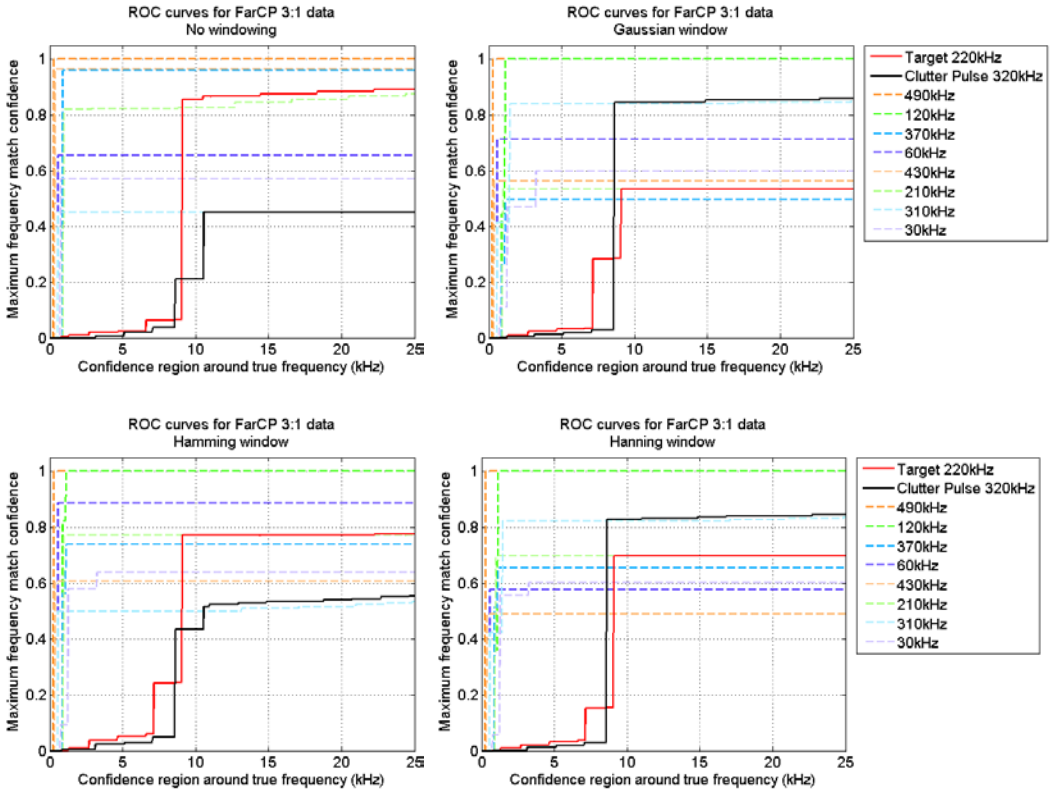


Figure 3.8: ROC plots for Far CP data with various pre-windowing functions. The CW components are arranged in the legend in decreasing order of their amplitudes.

As previously hinted by Figure 3.5, the chirp rate estimation remains overall very poor for the Far CP data case. Figure 3.9 shows the ROC curves calculated for the signed chirp rate of the returned atoms corresponding to target (solid lines) and clutter pulse (dashed lines), with and without windowing. The extent of chirp rate smearing (i.e., flat portions of the ROC curve) does not appear to change with different windowing functions. In terms of the chirp fidelity of the atoms, for the target signal the estimated chirp rate deteriorates with windowing; for the non-windowed case the best match confidence is still just above 0.6 and is obtained for a very large chirp rate confidence region, i.e., 15 kHz/s. The corresponding chirp rate for this largest confidence is now 30.5 kHz/s, that is, very close to the clutter pulse chirp rate. It is likely that atoms counted in this larger region have chirp rates corresponding to the clutter pulse (as seen in Figure 3.5), hence the jump in target confidence.

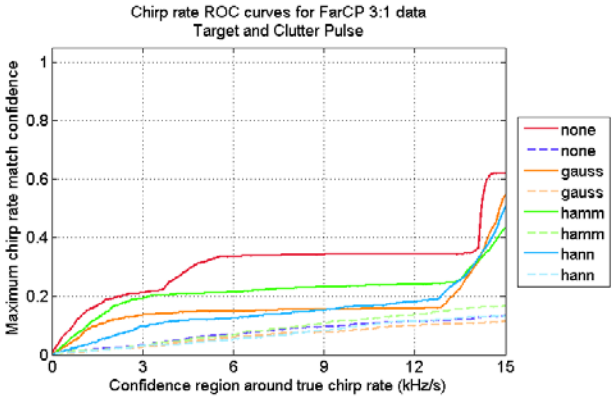


Figure 3.9: Chirp rate confidence for Far CP data with various pre-windowing functions. Both target chirp rate confidence (solid lines), and clutter pulse confidence (dashed lines) are poor.

Figures 3.3-3.9 show that preprocessing the data by windowing does not necessarily lead to higher fidelity of extracted atoms, for the test cases explored and for classification purposes. The windowing functions tested perform similarly and, depending on the exact input data and the metric used, they each can outperform the other by a small margin. Conversely, another conclusion is that the feature quality of the extracted atoms does not appear to be sensitive to the use of data windowing in the case of SNR 3:1, and remains generally as good or as poor as the non-windowed case. Windowing may still be useful for quality of input reconstruction purposes, but based on the study outlined above, it does not provide compelling advantages for classification.

3.2.2 Noise effects

Robustness to noise is highly desired in any data analysis application. For the simulated data introduced in Section 1.1, the signal-to-noise and signal-to-clutter ratios are relatively challenging, yet representative of a non-ideal, real application.

The reduced complexity test case of Section 3.2 is now used to evaluate the fidelity of returned atoms as the target amplitude progressively worsens, that is, the SNRs considered are 3:1, 1:1, and 0.3:1. The CW and pulsed clutter amplitudes remain unchanged. Figure 3.10 shows chirp rate versus frequency scatter plots of first 25 returned atoms, aggregated over ON windows in each SNR case. The OFF window case has similar characteristics and is not shown. Visually, it is readily apparent that the level of noise relative to the target signal directly impacts the order

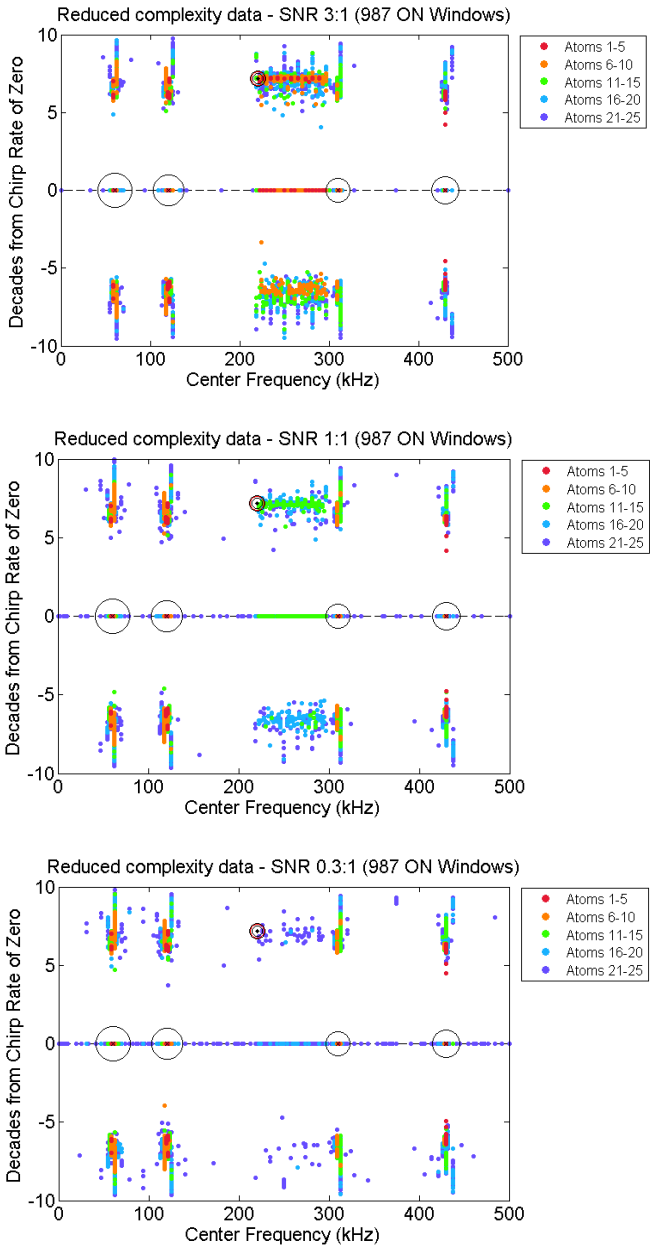


Figure 3.10: Scatter plots (chirp rate vs. frequency) for reduced complexity data for SNR 3:1 (top), SNR 1:1 (middle), and SNR 0.3:1 (bottom). Shown are the first 25 extracted atoms, aggregated over ON windows, and colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane. The target becomes less and less distinguishable as the SNR decreases.

of atom return, as expected. Also, the number of target-related atoms decreases significantly as the SNR decreases, and for SNR 0.3:1 the target becomes almost indistinguishable. In terms of the smearing effect for the chirp rates, it does not seem to be positively or adversely impacted by the level of noise. Since the impact of noise is obvious on the reduced complexity data, it is not relevant to further study the fidelity of the atoms and the corresponding ROC curves, but rather it is more useful to proceed toward analysis of Far CP data.

The Far CP data is now considered for the 3 SNR regimes, and Figure 3.11 shows scatter plots of chirp rate vs. frequency for the first 25 extracted atoms, colored according to their order of return and aggregated over ON windows for each SNR case. The impact of the reduction in SNR is severe, and in the 0.3:1 case there are no target-specific atoms returned by the pursuit search. It is possible that if more than 25 atoms were considered, some target atoms would be encountered. However, every additional atom comes with increased computational overhead, and the fidelity of any target atoms will very likely be poor. Also noticeable in Figure 3.11 is the lack of chirped clutter atoms, as the vertical atoms bins present in the respective area are likely due to the CW component at 310 kHz, based on the reduced data scatter plots of Figure 3.10. In terms of the degree of chirp rate smearing, no noticeable impact due to noise is observed, and the chirp rate estimation remains equally poor for the three SNR values considered. In the case of pre-windowing the data with a Gaussian window as specified in Section 3.2.1, the noise impact on the chirping pulses is mitigated slightly, as can be seen in Figure 3.12.

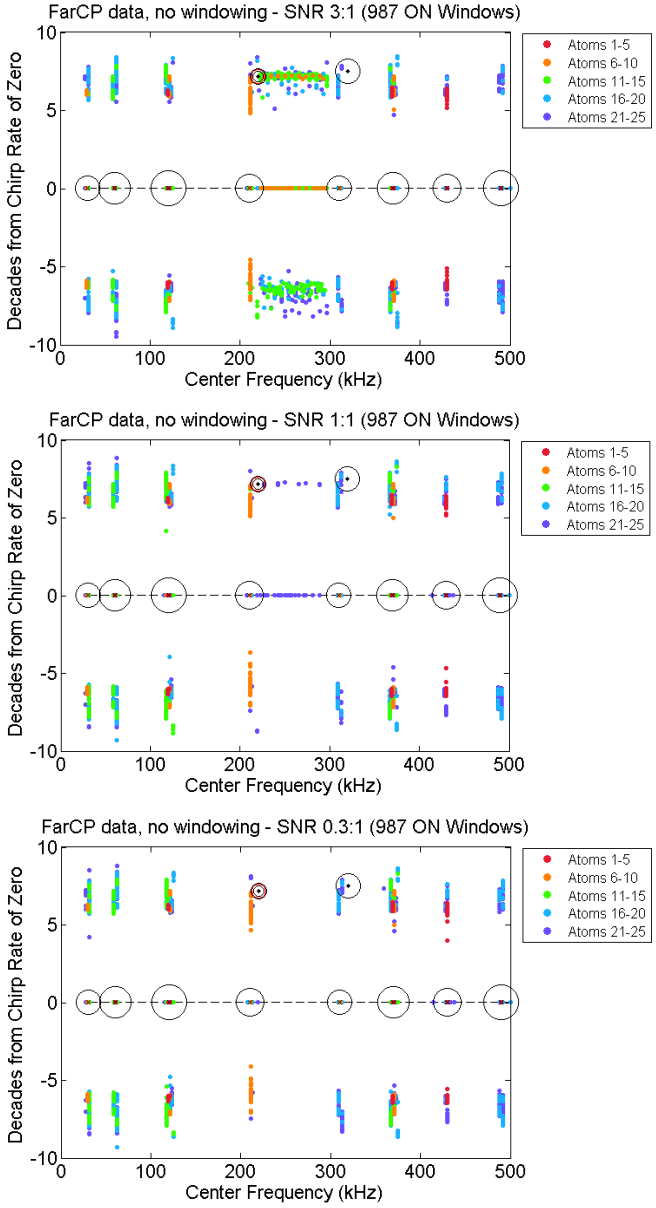


Figure 3.11: Scatter plots (chirp rate vs. frequency) for Far CP data for SNR 3:1 (top), SNR 1:1 (middle), and SNR 0.3:1 (bottom). Shown are the first 25 extracted atoms, aggregated over ON windows, and colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane. The noise impact on target atoms is severe.

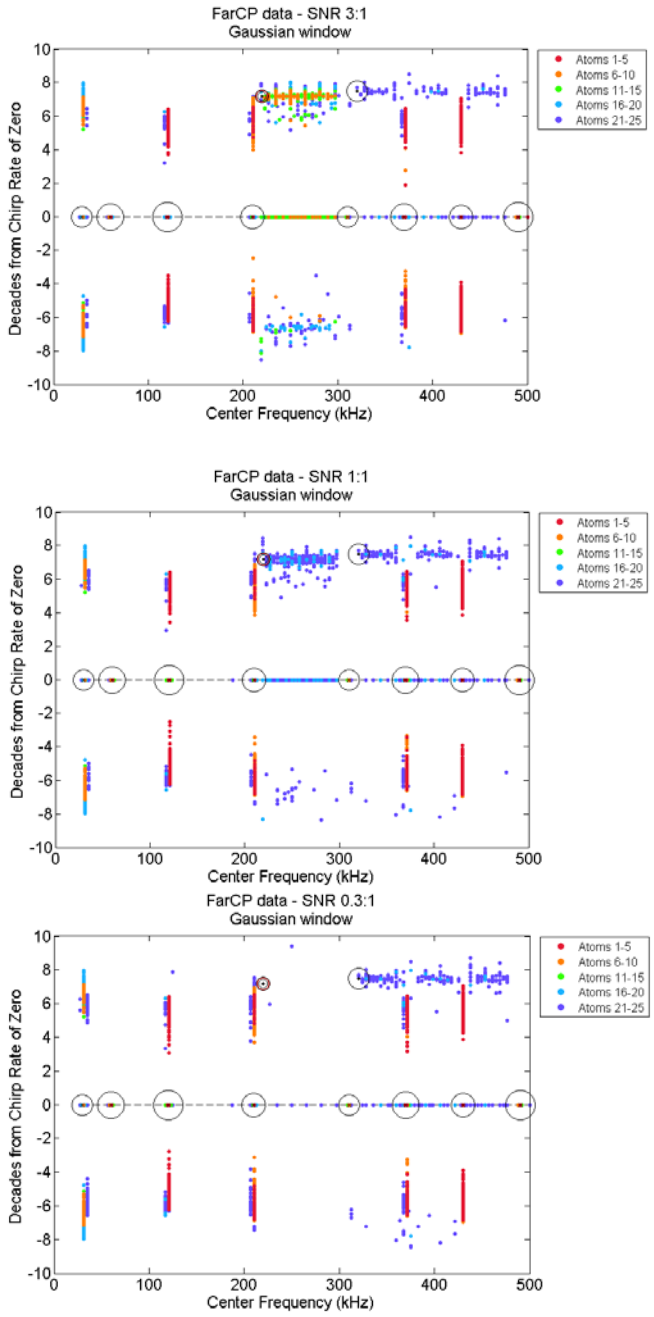


Figure 3.12: Scatter plots (chirp rate vs. frequency) for Far CP data for SNR 3:1 (top), SNR 1:1 (middle), and SNR 0.3:1 (bottom) with Gaussian pre-windowing. Gaussian pre-windowing slightly mitigates noise effects, and atoms corresponding to the competing chirped emitter now appear. Shown are the first 25 extracted atoms, aggregated over ON windows, and colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane.

Compared to Figure 3.11, chirped clutter atoms are now present among those returned by MP. Also, the number of returned target atoms increases, a fact that is especially obvious in the SNR 1:1 case. In the SNR 0.3:1 case, there are still a reduced number of chirping atoms in the target neighborhood. For the CW components, however, the impact of windowing on the fidelity of the returned atoms is negative. Whereas in Figure 3.11 the first returned atoms (red dots) were usually within the visual target perimeter, in Figure 3.12 chirp rate smearing is observed even for those first order atoms. One overall slightly positive effect of windowing is on the degree of chirp rate smearing, which is now reduced by few decades in all SNR cases.

Of the three SNR regimes, the most striking impact of windowing is seen for SNR 1:1 (compare middle panels of Figures 3.11 and 3.12). A closer look at the match confidences and the fidelity of the returned atoms for this noise case is afforded by Figure 3.13. The top panel of Figure 3.13 confirms that the first returned atoms for the CW components are no longer as closely matched to the true signals, as previously shown by the smearing of red dots in Figure 3.12. For the two chirping signals, windowing now leads to three dedicated returned atoms (top right panel), compared to only two dedicated returned atoms in the non-windowed case (top left panel). The fidelity of the atoms corresponding to some of the signal components is not as high in the windowed case as the non-windowed case (bottom panel).

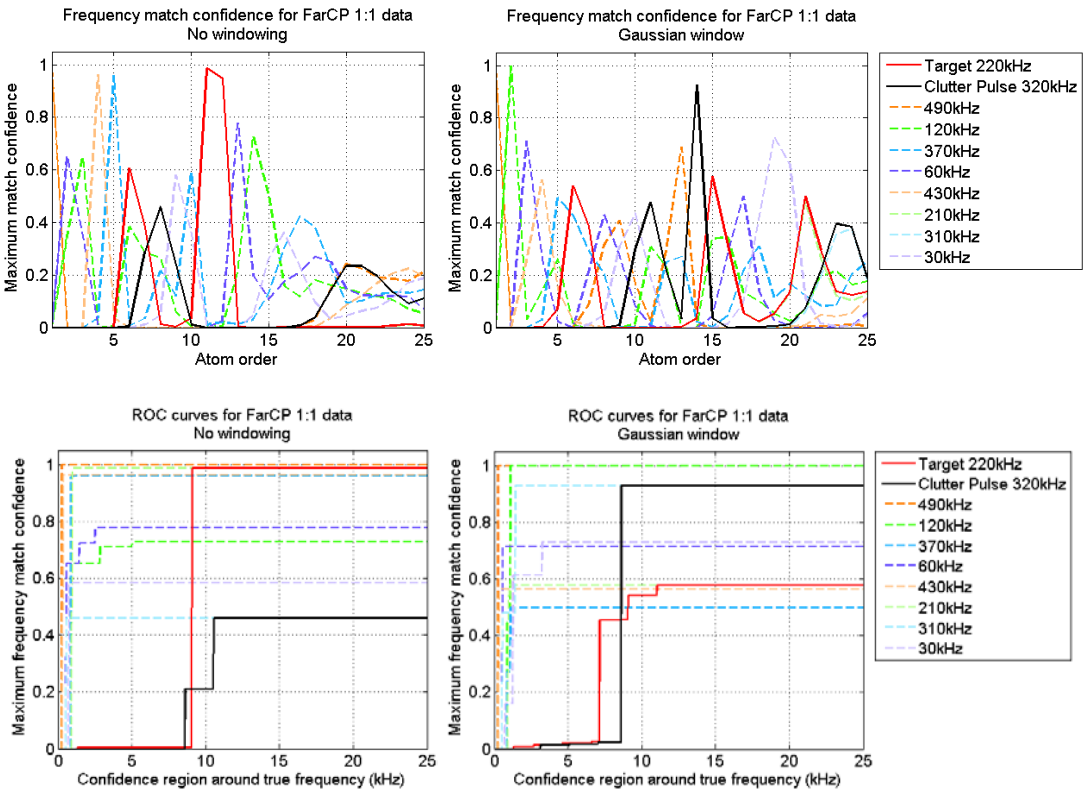


Figure 3.13: Match confidences (top panel) and ROC plots (lower panel) for Far CP SNR 1:1 data without windowing (left panels) and with Gaussian pre-windowing (right panels). Windowing the data appears to help improve the detection of chirping signals in very poor SNRs, but diminishes the detection fidelity for CW signals.

Figures 3.10-3.13 show that using overcomplete parametric dictionaries in conjunction with the fast ridge pursuit search is a noise-sensitive approach. It is possible that the method would perform much better in nominal SNR conditions of 10-20db, as indicated by the results in [120]. Windowing the data appears to help improve the detection of chirping signals in very poor SNRs, as shown by a representative example, but diminishes the detection fidelity for CW signals. These limitations will be summarized later in the text and will motivate the next steps of thesis work.

3.2.3 Dictionary overcompleteness

The chirped Gabor dictionary can be built with a range of resolutions for each of its generating function parameters, that is, its degree of overcompleteness is a user specification. In Gribonval's original formulation, the scale has a dyadic resolution (i.e., $s=2^j$), and the frequency resolution at each scale is then simply $d_f \leq 2\pi/s$. The chirp rate resolution in the fast ridge pursuit search is directly correlated to the scale and is equal to the frequency resolution, specifically, $d_c = d_f$. Thus, dictionary completeness plays a crucial role and must be looked at in greater detail to determine if perhaps by increasing the degree of completeness, higher parametric fidelity can be observed in the returned atoms, in particular for the chirp rate parameter.

The impact of the degree of dictionary overcompleteness is explored only in terms of the frequency (and thereby chirp rate) resolution. The scale and time centers are kept with the original resolutions specified in [120] (i.e., dyadic and unit step, respectively). An overcompleteness factor, o_f, is introduced for the frequency, f, resulting in $d_f=2\,\pi/(s*o_f)$ (and chirp rate resolution $d_c=d_f$).

The reduced complexity dataset is considered first, and the quality of returned atoms is assessed in terms of dictionary overcompleteness. Figure 3.14 below shows the frequency fidelity of the returned atoms for $o_f=\{1, 2, 4, 8\}$.

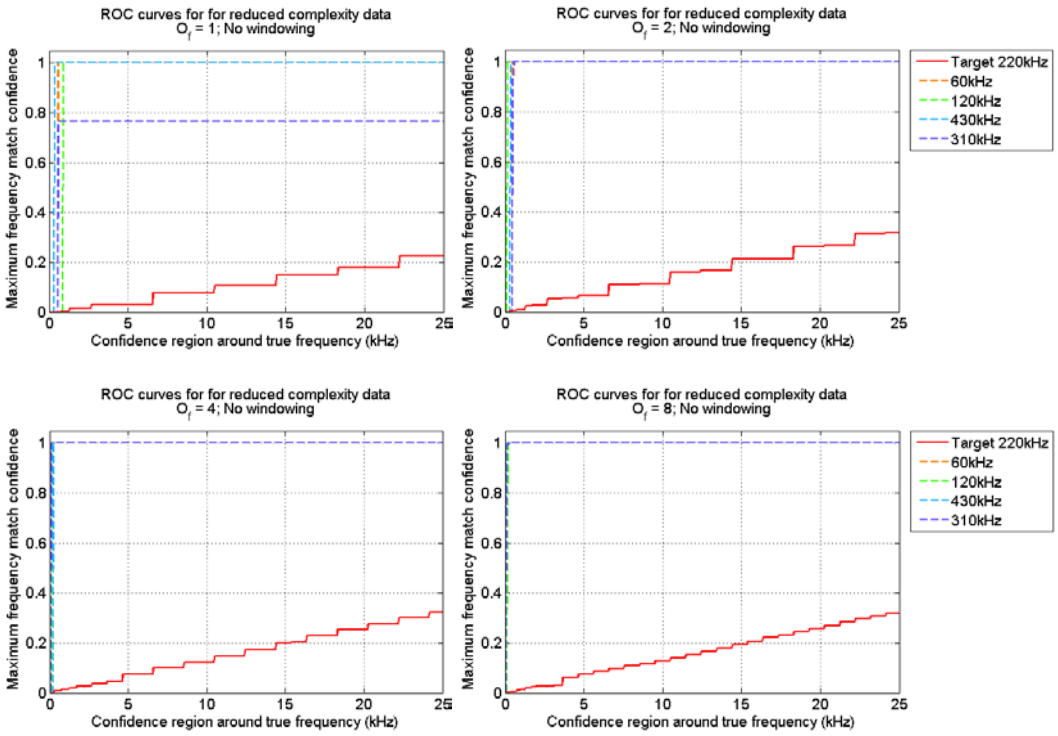


Figure 3.14: ROC plots for reduced complexity data for various degrees of dictionary overcompleteness.

The most significant improvement in the fidelity of the atoms for the reduced complexity data, both for the CW and the chirped signal components, occurs when the frequency oversampling is increased from o_f=1 to o_f=2. The improvement obtained by increasing the frequency resolution beyond o_f=2 is not as significant and comes at a higher computational cost for the dictionary search. Oversampling the frequency space implies a larger number of dictionary atoms is available for matching the input signal. This effect is illustrated in Figure 3.14 by the increasingly finer step in the confidence match for the target signal.

The Far CP SNR 3:1 data is now analyzed using dictionaries that are overcomplete by a factor of o_f={1,2}, with and without pre-windowing with a

Gaussian function. The fidelity of the returned atoms is shown in Figure 3.15 below. The left side plots of Figure 3.15 have been previously seen in Section 3.2.1 in the context of windowing effects. The highest atom fidelity for all signal components is now reached in the case of non-windowed data with o_f=2, similar to the reduced complexity case shown in Figure 3.14. When pre-windowing is applied in the o_f=2 case, the fidelity for some of the components decreases significantly from the non-windowed case, even though it is slightly higher for the target atoms. It is important at this point to also consider the match confidence for the four cases shown above as a function of returned atom order (Figure 3.16).

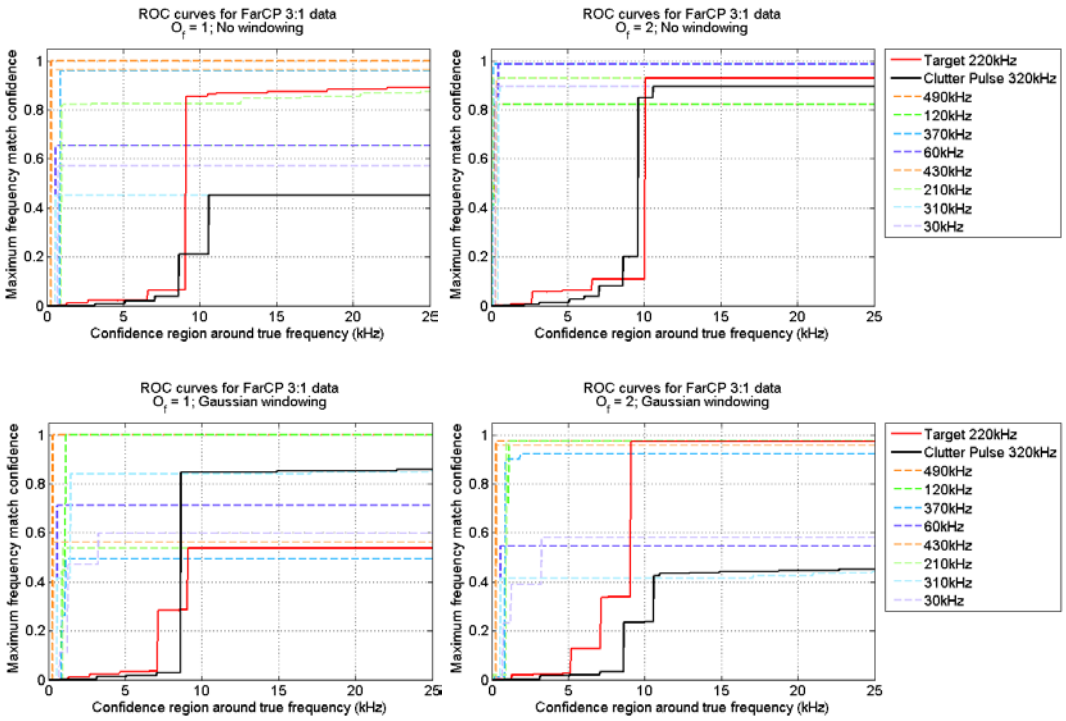


Figure 3.15: ROC plots for Far CP data for two degrees of dictionary overcompleteness, without windowing (top panel), and with Gaussian pre-windowing (bottom panel). Two overcompleteness cases are considered: o_f=1 (left panel), and o_f=2 (right panel). Higher fidelity is observed for the dictionary case with a higher degree of overcomplenetess.

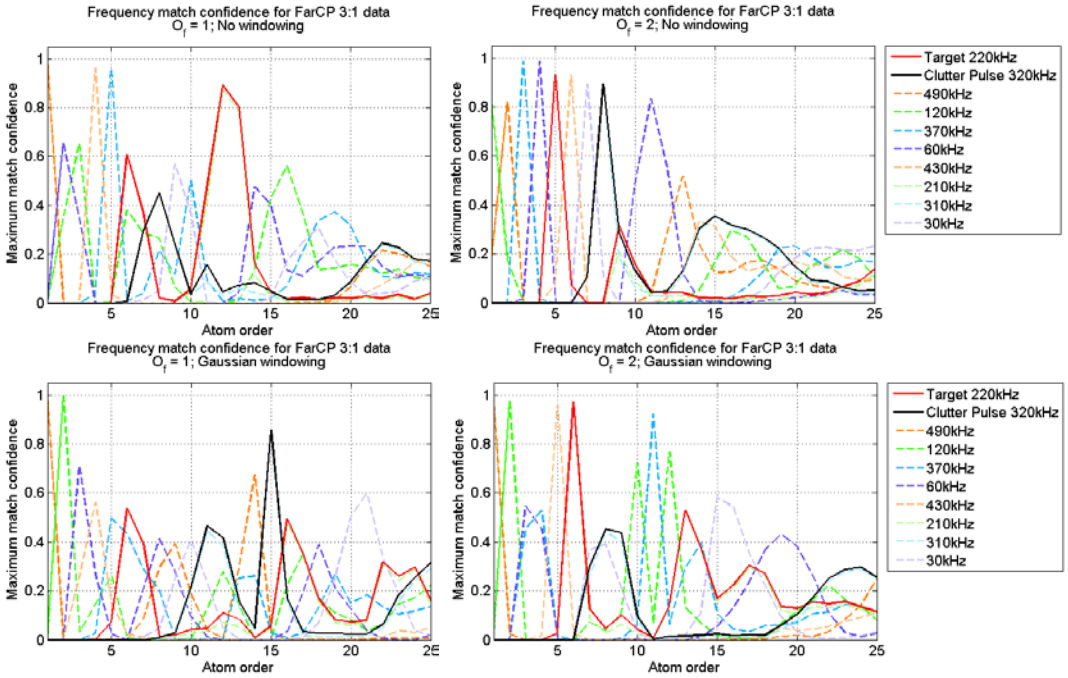


Figure 3.16: Match confidences for Far CP data for two degrees of dictionary overcompleteness, without windowing (top panel), and with Gaussian pre-windowing (bottom panel). Two overcompleteness cases are considered: o_f=1 (left panel), and o_f=2 (right panel).

Figure 3.16 strengthens the conclusion that the non-windowed case with a more overcomplete dictionary (top right panel) is better suited, among the cases tested, to extract higher quality atoms, both for the chirped components (i.e., target and clutter pulses), as well as most of the CW components. The frequency match confidences for the first 8 returned atoms is much higher in this case compared to the non-windowed o_f=1 case. Windowing in the o_f=2 case diminishes the positive impact of the increased dictionary frequency resolution.

The degree of dictionary overcompleteness impacts the quality of the returned atoms, both in terms of their frequency match confidence, as well as their respective order of return. Increasing the resolution of the parameter space, however, comes at

an additional computational cost for the dictionary search step, and the author notes it does not help improve the chirp rate smearing.

3.2.4 Clutter impact

Sensitivity to CW clutter is of practical importance, and is explored starting with a slightly modified Far CP timeseries, consisting of the target signal with SNR 3:1, the chirping clutter pulse, and one out-of-target-band CW emitter at 30 kHz (i.e., the first of the eight CW emitters in increasing frequency order). The number of CW components is progressively increased up to all eight CW signals in the Far CP data, with the remaining seven frequencies {60 kHz, 120 kHz, 210 kHz, 310 kHz, 370 kHz, 430 kHz, 490 kHz} added in order. Each timeseries is decomposed over the dictionary using 25 atoms and the resulting scatter plots of chirp rate vs. frequency are shown in Figure 3.17. For the cases with less than 3 CW components, many of the higher order atoms appear to be modeling the noise, as expected. As the number of individual signals in the timeseries increases, more and more atoms are used to capture the actual signals. The number of CW components appears to have no impact on the degree of chirp rate smearing. Given all the similar scatter plots observed in the previous sections, it would appear that the chirped signals in the timeseries perhaps induce matching chirp rates on the CW components, resulting in the erroneous atom matches. To test this hypothesis, the same 8 scatter plots are now shown in Figure 3.18 for a simple timeseries without any of the chirping components, i.e., the target and the chirped clutter pulse.

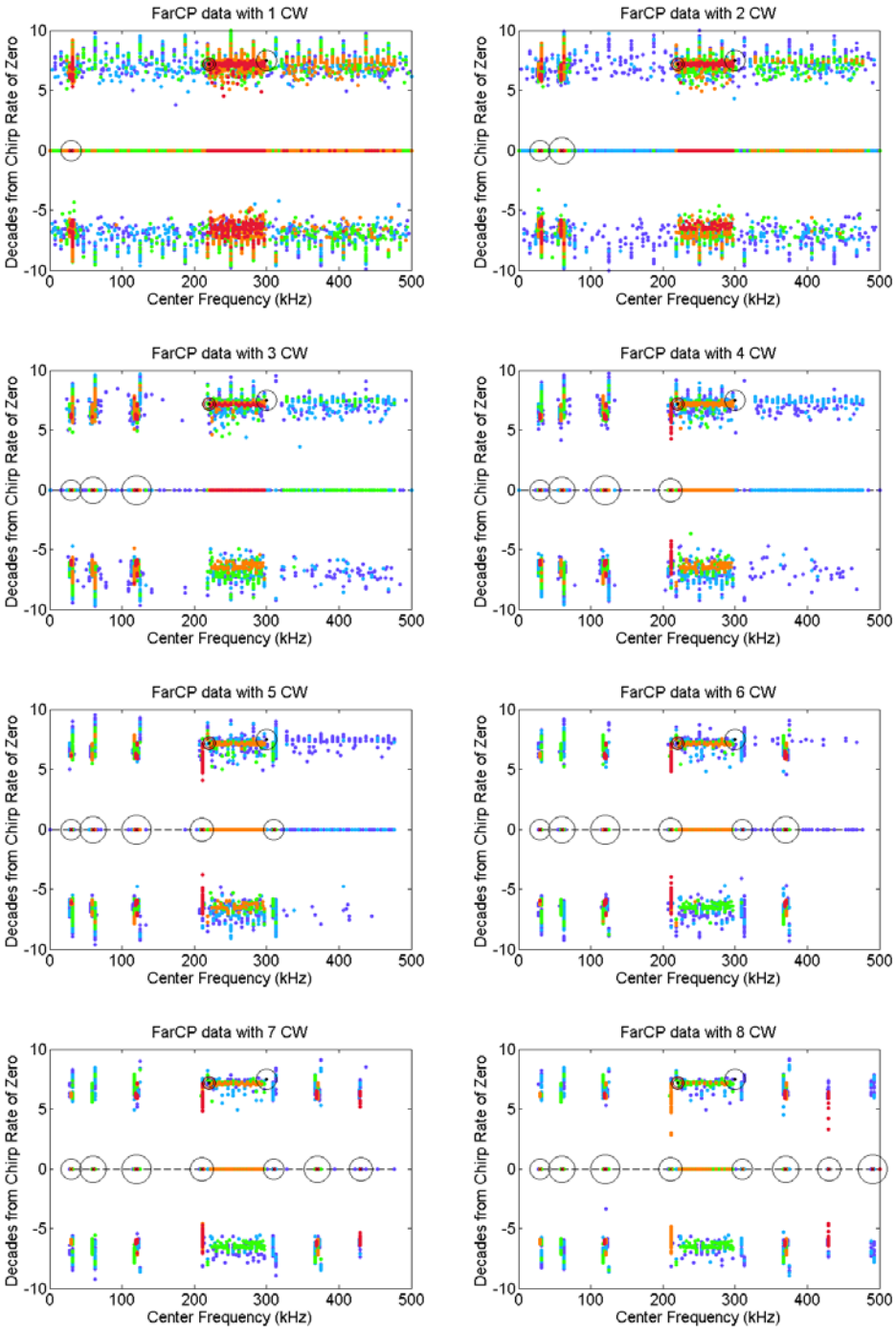


Figure 3.17: Scatter plots (chirp rate vs. frequency) for Far CP data as the number of CW components increases from 1 to 8. Shown are the first 25 extracted atoms, colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane.

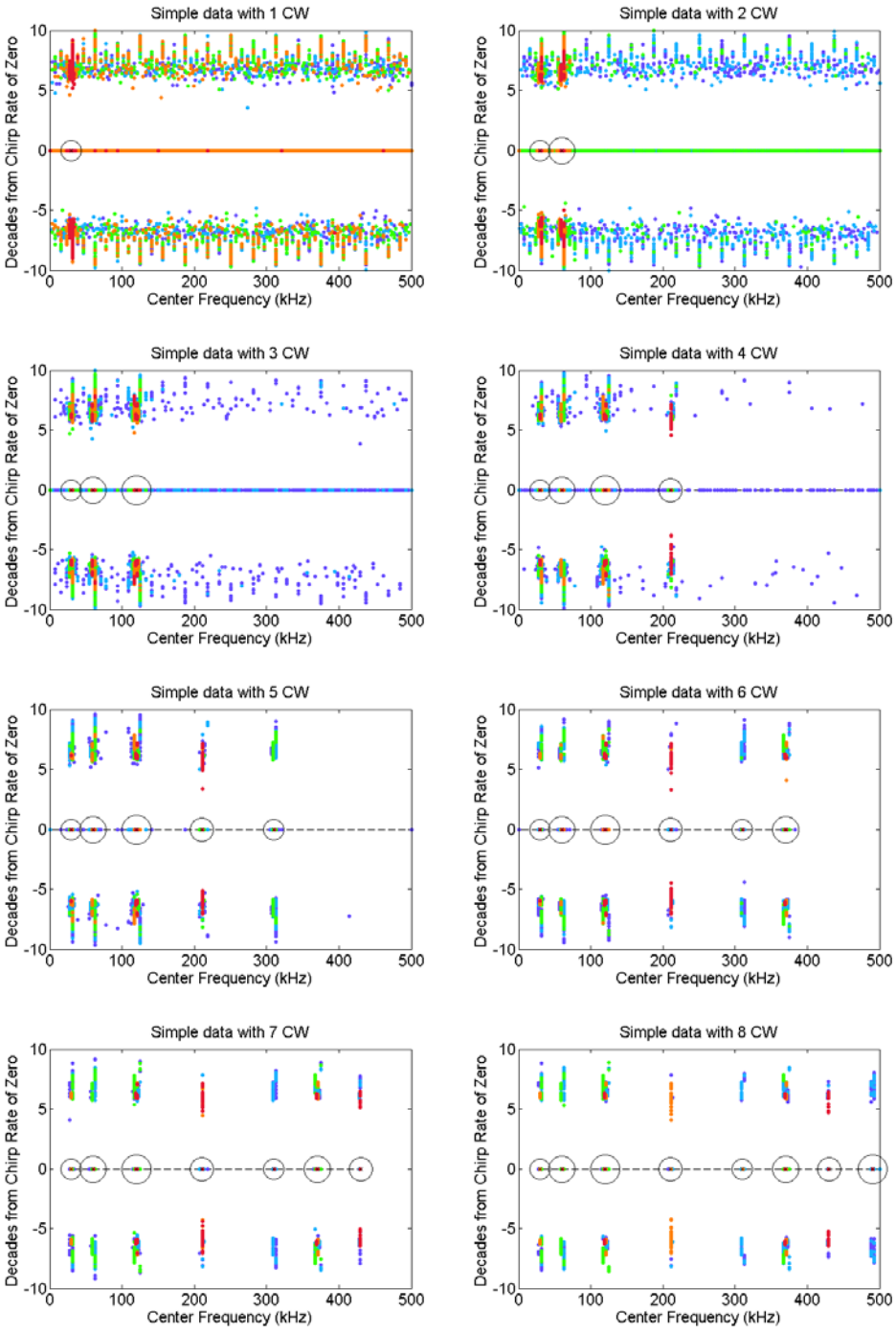


Figure 3.18: Scatter plots (chirp rate vs. frequency) for simple CW only data as the number of CW components increases from 1 to 8. Shown are the first 25 extracted atoms, colored according to their order of return by the pursuit search. The circles represent the locations of the true signal components in the frequency-chirp rate plane.

The same smearing effect is observed, and more importantly, there are still erroneous chirp rates greater than +/-5 decades, even though the timeseries contains no chirping components. Considering that this chirp rate error behavior was observed for all the different windowing, noise, and clutter scenarios explored, a natural conclusion is that this is a systematic algorithm error and is in fact due to the fast ridge pursuit search of the dictionary. The effect of the pursuit search can be observed in Figure 3.19 below. Shown here is the scatter plot of a frame operator for a window of the simple timeseries with 8 CW components and no chirping signals. The frame operator is simply the inner product of the input with all dictionary elements, ***without*** any matching pursuit subtraction. Figure 3.19 shows all the inner products for a sample window, plotted at coordinates equal to the corresponding atom frequency and chirp rate, and colored according to the particular inner product weight (i.e., the largest inner products are in red and the smallest in blue). The *y*-axis range is the same as that of Figures 3.17-3.18 for easy comparison.

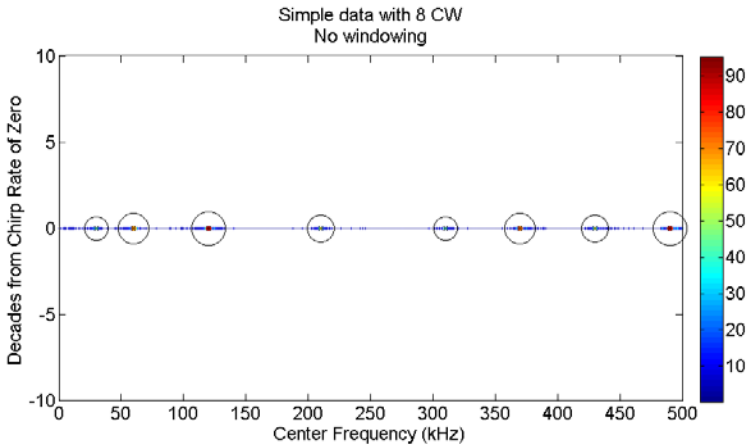


Figure 3.19: Scatter plots (chirp rate vs. frequency) for simple 8 CW data. Shown are all the dictionary atoms without matching pursuit subtraction, colored according to their respective inner product weight. The circles represent the locations of the true CW signal components in the frequency-chirp rate plane.

There is no chirp rate smearing effect in Figure 3.19 and the returned top atoms correctly match the true signal components. What this means is that the fast ridge pursuit indeed impacts the parametric fidelity of the returned atoms. One possible explanation is the use of ‘dual molecules’ in the fast ridge pursuit of [120], which mathematically serve the purpose of converting complex atoms to real ones. These dual molecules use conjugate complex atoms and could perhaps explain the dual chirp rate behavior in the atoms. Somewhat similar behavior of increasing and decreasing instantaneous frequencies was remarked upon in [7, 120], but was considered to be a strength of the algorithm in decomposing a vibrato acoustic signal with very few chirps. However, the extensive study in this chapter shows the dual chirp rate behavior and the chirp rate smearing is more a function of the fast ridge pursuit than the particular signal to be decomposed. From a feature extraction perspective, if chirp rate is a feature of interest, this is a weakness that renders Gribonval’s pursuit method unusable.

3.3 Dictionary representation fidelity

A key observation is that poor parametric fidelity of the returned atoms does not imply poor reconstruction fidelity of the input signal. In [7, 120], the performance metric used was representation fidelity of the input, whereas in this work the metric is parametric fidelity (i.e., recognition accuracy). The chirped Gabor dictionary can in fact be efficient in representing the simulated data, and Figure 3.20 illustrates this property. The semilog plot of Figure 3.20 shows the average normalized residual after

every matching pursuit iteration, that is, as more dictionary elements are added to the representation. This residual can also be thought of as reconstruction error at that particular iteration. The Far CP 3:1 data is considered in this instance, and the average is calculated over all the 987 ON windows in a timeseries. Several different dictionaries are compared in terms of their sparse representation capability.

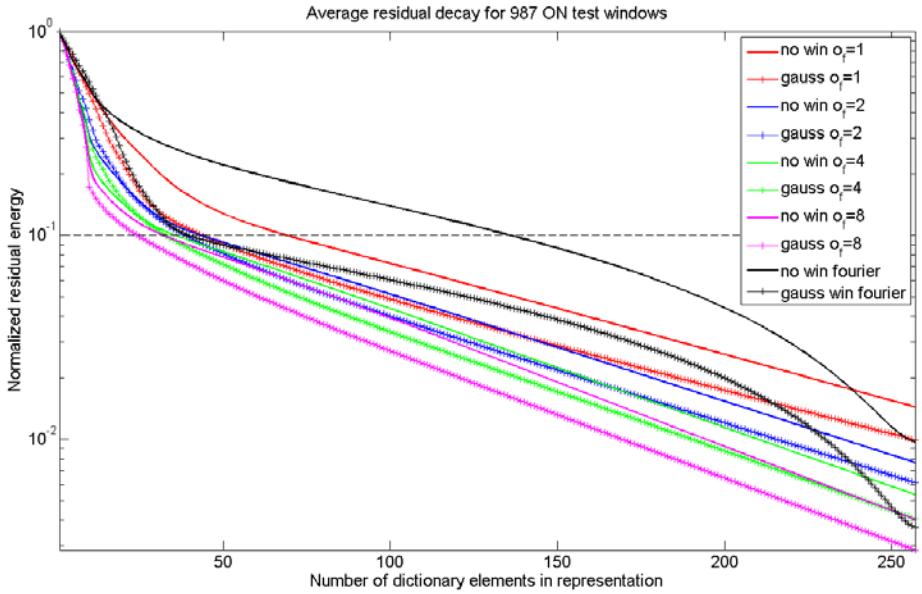


Figure 3.20: Residual decay averaged over all ON windows for various analytical dictionaries, with and without Gaussian pre-windowing.

First, it is observed that there are eight chirped Gabor dictionaries with several degrees of overcompleteness, each with and without Gaussian pre-windowing (colored traces). Secondly, a Fourier dictionary is considered, with and without pre-windowing (black traces). The Fourier dictionary is in fact the N=512-point FFT basis used with a matching pursuit decomposition instead of the usual projection. Since it is orthonormal and the projection of a given input is unique, the matching pursuit

decomposition will in fact replicate the contribution of the FFT coefficients, sorted according to their amplitude. Only the first 256 decompositions are considered in Figure 3.20 (i.e., one-sided frequency spectrum).

Good sparse approximations capture most of the signal energy in the first few coefficients, which translates into a very steep initial decay of the residual. In this sense, the more overcomplete the Gabor dictionary, the more energy compacting it is (compare residual decay for red, blue, green, and magenta traces). All the overcomplete dictionaries outperform the Fourier dictionary case without pre-windowing. It is interesting to note that for the first ~12 dictionary elements the Fourier dictionary and the standard overcomplete chirped Gabor (o_f=1) have very similar residual behaviors, with or without windowing. This could be due to the fact that the smallest frequency resolution in the dictionary at o_f=1 matches that of the Fourier dictionary, and the initial selected atoms correspond to CW components, i.e., they have zero chirp rates, and therefore are very similar to the selected Fourier elements of the same CW frequencies.

The efficiency of the dictionaries in Figure 3.20 in capturing the input signal energy can also be evaluated by counting how many atoms are needed to represent 99% of the signal (dashed black line). **The overcomplete (o_f=8) dictionary with pre-windowing needs ~25 atoms to reach the 99% threshold, while the Fourier dictionary needs ~136 elements to capture the same amount of energy.**

In terms of reconstruction error, the more overcomplete the Gabor dictionary, the better (i.e., lower) the reconstruction error, which is a slightly different effect than

was observed on the fidelity of the returned atoms. Also, pre-windowing the data in general improves reconstruction performance, and the trend for every dictionary considered is for the windowed case (cross traces) to yield lower residuals than the non-windowed case (flat traces) after the first few extracted atoms. This is also different from what was observed for the atom fidelity performance, as windowing was observed to hinder feature extraction.

3.4 Software implementation and algorithm complexity

The dominant challenge for RF signal processing is the length of the time records, the high sample rate, and the equivalent short processing time available for real-life applications. The non-chirped Gabor overcomplete dictionary is pre-generated using a fast numerical implementation and subsequently used in tabulated format. For the modeled data rate of 1 MHz, with an analysis window of 512-sample length, the non-chirped dictionary size is 5120 atoms for overcompleteness factor o_f=1. Given a timeseries with 500,000 samples (i.e., 0.5 s at 1 MHz sampling rate), the equivalent real-time processing for classification of all the resulting time analysis windows is 0.5 s. The original fast ridge pursuit algorithm was implemented in Matlab, and even though it was much faster than a brute force search of the entire Gabor dictionary based on the comparison in [120], it still resulted in run times few orders of magnitude higher than real time. Computational speedups were achieved by using parallel processing whenever possible, by vectorizing the dictionary search, and by extensive use of logical masks. The decomposition of buffered test data (i.e.,

matrix of all overlapping analysis windows in a timeseries) over the dictionary using the optimized implementation has a computational time within one order of magnitude of real time, on a Windows workstation with 8 Intel Xeon X5550 2.67GHz quadcore processors. The search for one atom is ~4 s for the entire buffered timeseries of 1952 analysis windows, and ~0.44 s for a single analysis window, on the specified architecture. Even though the complexity of the fast ridge pursuit search is only order *O(NM)* [7], for the dictionary size employed it is not achieving real-time atom extraction, despite the many optimizations. Additional gains in speed were achieved by distributing the calculation of inner products with the dictionary across the cores of a graphical processor unit (GPU). Even so, the feature extraction step alone takes longer than the real-time cutoff, without including the additional classification step, which makes the entire overcomplete, parametric dictionary classification approach impractical for real-time implementation.

3.5 Conclusion on overcomplete parametric dictionaries

The fast ridge pursuit in a chirped Gabor dictionary is fairly accurate in extracting the frequency of the signal components present in the test data. However, significant errors in the estimated chirp rate parameter limits the use of chirp rate as a feature; within a frequency band, the chirp rates vary by several orders of magnitude. Signals in distinct frequency bands can be distinguished from one another, while signals in the same frequency band with distinct chirp rates cannot be distinguished

from each other. In general, this is a significant problem and precludes the use of this method as the basis of a robust classification scheme.

Another downside is that an increasingly large number of dictionary elements are required to capture the target as the signal becomes dominated by clutter sources. As described in Section 3.1.1, matching pursuit identifies the dictionary element that best captures the signal at the current iteration, i.e., minimized local residual as opposed to global residual. Thus earlier elements capture the stronger signals, and therefore capture noise and clutter before finding a quiet target. As a result, the number of elements chosen to represent the data becomes crucial. If too few elements are used (e.g., 5 for Far CP data – red dots in scatter plots), the target never appears in the selected elements. If too many are used (e.g., 100), the number of false positives incurred becomes debilitating; the more dictionary elements we consider, the more likely we are to observe a spurious match to something approximating the target characteristics.

The accuracy and reproducibility of the parameters obtained using the chirped Gabor dictionary with the fast ridge/matching pursuit search was found to be sensitive to the amount of clutter, SNR, overcompleteness, and the choice of data windowing function (e.g., rectangular window vs. Hamming window). This sensitivity is partly due to the fast ridge dictionary search and the matching pursuit greedy approximation. Raising the degree of dictionary overcompleteness by increasing the frequency and chirp rate resolution can lead to better parameter matches, but the associated computational overhead is very large. Improvements in feature quality

could be made by using the frequency information of the returned atoms, which was the most accurate of all estimated parameters. For example, the target chirp rate could be estimated by considering all the returned frequencies in the target spectrum and fitting an instantaneous frequency line to those observations, whose slope would be the chirp rate. This approach presents its own challenges and would only incur additional computational time.

The purpose of this chapter was to rigorously explore the parametric overcomplete dictionary approach and focus on the quality of the features extracted. The effects on the atoms' parametric fidelity introduced by windowing, reduction in SNR, amount of clutter, and dictionary overcompleteness were evaluated and the approach was found to be sensitive to all of them. The conclusion of this section is that for the specific RF application modeled, this analytical dictionary approach was found to be unreliable for classification, and nearly impossible to implement for real-time classification. It provided, however, one very important insight into dictionary methods in classification: a dictionary with good reconstructive properties may not necessarily provide good classification features. This key finding motivates the alternative approaches explored and developed in subsequent chapters of this thesis.

4. Sparse Representations in Learned Dictionaries

In this chapter it will be shown that dictionaries learned directly from the data can eliminate the need for prior knowledge of clutter or target characteristic models, lead to sparse representations, and perform well in conjunction with a statistical classifier. At this point in the study, the data is split into two disjoint sets, one for training, and one for testing, to ensure proper validation of results. Training data and test data consist of distinct sets of timeseries for each of the three cases (Far CP, Flat CP and Chirped CP), where every individual timeseries is generated with different initial conditions (e.g., random noise seeds). In the sections that follow, the dictionaries are always learned from the training sets, and the classification accuracy is evaluated on the test sets.

The process is as follows: first, a modified on-line batch Hebbian learning algorithm similar to [18, 91, 96] is used to learn dictionaries for RF signal classification from training data. The sparse approximations of test data obtained via matching pursuit are considered to be “features” and are used in conjunction with a minimum residual (MR) classifier (or nearest subspace) [15, 97], to distinguish between the time windows when the target signal is ON or OFF. Secondly, the K-SVD method of Aharon et al. [10] is also implemented to build classification dictionaries from the same training data, and similarly used with the MR classifier in order to compare the performance of the two dictionary learning methods. The K-

SVD algorithm is similar to the K-means clustering process, and it works with any form of sparse signal representation algorithm. It is an established approach, and is more similar to vector quantization, whereas Hebbian learning was developed as a neuromimetic learning technique [92, 121]. The work in this thesis is the first attempt to extend and adapt dictionary learning methods to RF processing. Additionally, detailed comparison of two different learning algorithms helps generalize the results of this study to arrive at novel and applicable conclusions. These specific learning algorithms are detailed in Section 4.1.

In order to use learned dictionaries for RF signal classification, an analysis of the quality (e.g., specificity or uniqueness) of resulting features must be made from a discrimination point of view. Dictionary learning algorithms have a number of parameters that need to be optimally chosen under some metric, which in this work is *classification performance*. Selecting a dictionary size effective for classification is the first task, as this choice has significant impact on the computational demands, both from the standpoint of building the dictionary as well as extracting classification features. A large portion of dictionary design work in image processing has focused on very large (i.e., overcomplete) dictionaries for increasingly sparse representations. In recently published work [16], the author showed that, for different RF simulated data, dictionaries that were undercomplete by a factor of 20 lead to good classification performance in almost real time. It was the undercompleteness of the dictionary in particular that made near-real time classification feasible. Although intuitively one cannot expect great reconstructive performance from an

undercomplete dictionary, this research direction was motivated by the results in Chapter 3. There it was shown that excellent reconstruction capability did not lead to or correlate with good classification features. Conversely, if perfect reconstruction is not needed for perfect classification, there is no compelling reason to learn overcomplete dictionaries that can reconstruct well.

The impact of dictionary size is now explored in greater detail by examining performance for a range of dictionaries from undercomplete (by a factor of 32) to overcomplete (by a factor of 2). This chapter will show that good classification accuracy can be obtained with dictionaries that are undercomplete with respect to the length of the input vectors (i.e., natural dimensionality). Additionally, a sensitivity analysis on the learning algorithm parameters will be made with respect to classification accuracy.

4.1 Learning algorithms

An overview of the dictionary learning algorithms is now briefly presented. Given a signal set X containing P normalized training vectors x_i, each of length N, the dictionary learning process begins by initializing the K elements of dictionary Φ with l^2 normalized rows of random numbers from a uniform distribution. A more detailed discussion on the dictionary initialization is deferred to Section 4.1.1.

Learning Φ takes place over multiple iterations (the number of times the dictionary "sees" the entire training data set), C, and generally consists of two stages per learning iteration: the *sparse coding stage*, and the *update stage*.

In the *sparse coding stage*, which is the same for both Hebbian and K-SVD learning, a weight vector a_i is found for each training vector x_i using the current dictionary iteration $\Phi^{(c)}$, such that a_i is sparse and $a_i^T\Phi^{(c)}$ is a sufficiently good approximation to the input,

$$\min_{a_i}\left\{\left\|x_i - a_i^T\Phi^{(c)}\right\|_2^2\right\} \quad \text{such that } \|a_i\|_0 \le L, \tag{4.1}$$

where the sparsity factor, L, controls how many dictionary elements are allowed to represent a particular training vector. The sparse approximations that were used throughout this thesis were found using an l_0 "norm." As mentioned before, this problem is NP-hard, lacking an exact solution, but an approximate solution for a_i can be found using a simple matching pursuit algorithm [6]. This choice yields an easier, faster implementation, as well as a progressive way of increasing the sparsity of a particular approximation to study effects on classification performance.

Once a sparse approximation vector a_i is found, the dictionary learning proceeds to the *update stage*, which will be separately discussed for the two methods in sections 4.1.2 and 4.1.3.

The learning iterations, each with a sparse coding and an update stage, continue until some criterion is fulfilled. This criterion can be a measure of dictionary learning convergence (i.e., the individual dictionary elements stop changing significantly between consecutive updates), a measure of representative or discriminative power [14], or an empirically chosen fixed number C of learning

iterations. In this work, a range of fixed learning iterations C is considered in order to explore the exact behavior of dictionary learning convergence.

4.1.1 Dictionary initialization

Several recent publications have explored initialization with random noise vectors [10], by imprinting with actual data vectors [111], or by seeding the dictionary with a sparsifying transform on the data [73]. While there are many ways to initialize a dictionary, in this work all the elements are usually initialized with random normalized vectors, using distinct random seeds in each case, for the purpose of exploring how the dictionaries learn from the training data. Starting with a random dictionary, the first batch of training vectors will each "select" the best L dictionary elements for their sparse representations with relatively equal probability. These selected elements are updated and their likelihood of getting selected for representing the following batch is now increased. As the dictionary receives sequential training batches, additional dictionary elements are activated (i.e., become updated) to capture the remaining variability in the data.

4.1.2 Hebbian dictionary update

In the Hebbian learning case, the training set is viewed one training vector at a time (i.e., sequentially), and an update of the entire dictionary is performed in parallel for all dictionary elements, φ_k, for each training vector x_i using the learning rule:

$$\forall \varphi_k^{(c)} \in \Phi^{(c)},\ \hat{\varphi}_k^{(c+1)} = \varphi_k^{(c)} + \eta \Delta \varphi_k^{(c)}. \tag{4.2}$$

Here η is a parameter controlling the learning rate, and the new estimate $\hat{\varphi}_k^{(c+1)}$ is re-normalized to unit norm ($\left\| \hat{\varphi}_k^{(c+1)} \right\|_2 = 1$). That is, in each learning iteration the dictionary is updated as many times as there are training vectors x_i. The training vectors are received in a random order which changes at each learning iteration.

The dictionary element update, $\Delta\varphi_k^{(c)}$, is derived to minimize the energy cost function in (4.3) with respect to each dictionary element, for a known sparse representation of the input vector [92]:

$$E = \left\| x_i - \Phi^{(c)} a_i \right\|_2^2 + \lambda \left\| a_i \right\|_0 . \tag{4.3}$$

The first term measures how well the dictionary describes the training vector x_i, according to mean square error, while the second term enforces sparsity in the weight vector a_i via the sparsity constraint λ. The dictionary update in equation (4.4) is then obtained by performing gradient descent on this cost function with respect to the dictionary elements, resulting in:

$$\Delta\varphi_k^{(c)} = -2a_{i,k}(x_i - \Phi^{(c)} a_i). \tag{4.4}$$

A variation to this serial algorithm is a batch Hebbian learning, which allows small groups (i.e., batches) of I training vectors to contribute simultaneously to the update in equation (4.4), resulting in the batch update equation:

$$\Delta\varphi_k = \frac{1}{I} \sum_i 2a_{i,k}(x_i - \Phi a_i). \tag{4.5}$$

4.1.3 K-SVD dictionary update

For K-SVD learning, first a sparse matrix of weights, *A*, is found over the current dictionary iteration for all training data. Then, the *dictionary update stage* begins and in the K-SVD case, the training vectors are viewed simultaneously by the dictionary as a matrix *X*, and each dictionary element $\varphi_k^{(c)}$ is updated sequentially based on the group of training vectors it helps represent, similar to [10]. Let

$$R_k = X - \sum_{j \neq k} a_{j,*}^T \varphi_j^{(c)} \quad \text{and} \quad R_{k \in S} = \left\{ R_k^{i\ rows} \middle| a_{i,k} \neq 0 \right\}, \tag{4.6}$$

where the matrix R_k is the signal residual after the contribution of all dictionary elements different from $\varphi_k^{(c)}$ is subtracted from the signal matrix, *X*. The residual matrix is then restricted to rows $R_{k \in S}$ that represent the residuals for the training vectors that contain $\varphi_k^{(c)}$ in their sparse representation. Given the singular value decomposition

$$\left(R_{k \in S} \right)^T = U \Sigma V^T, \tag{4.7}$$

the dictionary update rule is $\varphi_k^{(c+1)} = \mathrm{u}_1^T$, where u_1 is the largest singular vector.

4.2 RF classification with learned dictionaries

In a typical classification setting, a user has to jointly optimize feature extraction and train the classifier, as shown in Figure 2.1. In Chapter 3 such a joint approach was unsuccessful with the use of a *parametric overcomplete* dictionary (as opposed to the *learned* dictionary introduced here), that would automatically provide the features a domain expert would use to categorize the simulated data in Figure 1.2

(e.g., frequency, chirp rate). Learned dictionaries constructed as outlined in Section 4.1 effectively learn features intrinsic to the data that allow direct classification, without necessitating a domain expert's input (e.g., a human). It is worth noting, however, that a domain expert is required to produce or select training data that is deemed meaningful for a given application.

In the context of RF data, one can view learned dictionary elements as a set of learned matched filters for the non-stationary training data. There is no longer a need for a domain expert to analyze and extract possible features and separately train a classifier on those features. The dictionary elements themselves become the features, and a match to those features (i.e., high degree of correlation) will tend to indicate the correct class. The hypothesis here is that the learned dictionary defines a space that is closely matched to that of the input data (i.e., matched space), allowing for good classification for a variety of learning parameters. In particular, when the parameter in question is dictionary size, the matched space can represent a dimensionality reduction for undercomplete dictionaries.

In order to use learned dictionaries on the simulated RF data of Chapter 1, the classification is posed as a two-class problem: discriminating between target ON and target OFF in a test data window. Therefore, the dictionaries are learned in pairs: one ON dictionary, one OFF dictionary. To classify a test timeseries, it is decomposed into length-N data vectors by using a sliding overlapping window. Two sparse representations of the signal in each window are then constructed via matching pursuit, one using elements from the ON dictionary, one using those from the OFF

dictionary. The MR classifier [15, 16] is then used to assign to the window the label corresponding to the dictionary yielding the smallest matching pursuit residual energy. That is, the MR classifier decides based on the best matched space, or the "nearest" space, to the input data.

Both training and test data sets are processed with data analysis windows of length 512 samples (0.5 ms of recording), with overlap of 256 samples, resulting in 1952 windows per timeseries. A true (ground truth) label is given to each data window using prior knowledge of the operational state (i.e., ON or OFF) of the target. A window is classified as an "ON-window" if the target pulse is present in 100% of the analysis window. A window is labeled as an "OFF-window" if the target pulse is completely absent. Windows containing partial target signal are ignored in this work.

The training set consists of two simulated time-recordings, i.e., a total of 1 s of data for training, from which an equal number of ON and OFF windows is selected. The resulting ON training data set consists of 1700 fully ON data vectors, and similarly the OFF training set includes 1700 fully OFF data vectors.

The test data consist of timeseries recordings of 0.5 s each, different from the training data. The goal is to classify the operational state of the target as recorded in the test data in each of the data windows. Each test time window is given an ON/OFF label by the dictionary pair via the MR classifier. These classification labels are compared with the true labels to obtain classification *accuracy* ((True ONs+True OFFs) / (Total ONs+OFFs)), which is used as the performance metric throughout this thesis.

To better evaluate the learning convergence properties of the algorithms, dictionaries are learned in sets of ten pairs – one ON dictionary, one OFF dictionary per pair – from training data with high amplitude (SNR 3:1) target, as shown in Figure 1.2. Each dictionary in the set of 10 pairs is initialized with a different random seed to eliminate the possible effects of correlated initialization, and to explore variability in learning behavior. Even though Hebbian dictionaries are the primary focus of this work, separate dictionary sets were learned with both K-SVD and Hebbian methods, with various parameter values for the sparsity factor, L, the dictionary size, K, and the number of learning iterations, C. These sets of dictionaries will be used to evaluate the classification accuracy convergence for the various parameters, and the variance of the performance across each set of 10 dictionaries. Classification convergence that is independent of dictionary initialization (i.e., exhibits little variation across the 10 pair set), is indicative of a useful learned dictionary for the type of signal represented in the training and test data sets. This variance is introduced by the random initialization of each dictionary prior to the learning process, and the goal is to see the dictionaries converge to similar performance regardless of the random seed (i.e., see the variance across the set decrease).

4.3 Dimensionality of training data space

While the amount of training data available varies from one application to another, it is always limited by practical constraints. On one hand the goal is to

minimize computational overhead in feature extraction and training a classifier, on the other hand the aim is to obtain good features for high performance classification. One factor that impacts this dual optimization problem is the size of the training set. Determining the minimum amount of training data required to obtain good classification is an ongoing research question in the field. While this work does not explicitly explore performance with respect to training set size, it is implicitly explored in the study of learning iterations, *C*. The value of *C* is a direct measure of how much training a dictionary goes through, and this chapter will demonstrate that by increasing the value of *C*, classification performance can be much improved, even for a training set as small as that considered here (i.e., only 1 s of data).

At this point, it is useful to explicitly introduce the concept of 'data space' in order to accurately describe training data used in this project. For a specific class of RF signals (e.g., those from arcs for example), there are likely many unique time domain measurements which each differ one from the other, but share similarities which distinguish them from other types of signals (e.g., pulsed CW waveforms, for example). Conceptually speaking, the set of all possible measurement outcomes of a given signal type constitutes what is termed 'data space.' In this sense, training data are all sampled from the same data space, i.e., from the same data distribution. It is this very data space the proposed dictionary learning methods are aiming to learn from the training data. The training set therefore must include sufficient information (i.e., entropy) to adequately describe the entire data space. The algorithms studied

here also rely on multiple learning iterations, that is, they are allowed to view the training data multiple times to improve their approximation of the data space.

One way to visually explore the data space is to perform principal component anaylsis (PCA), and look at the principal components (eigenvectors) of the matrix formed with all the training vectors. Figure 4.1 shows spectrograms of the first 20 principal components (i.e., those with the largest 20 associated eigenvalues), for the ON (top panel) and OFF (lower panel) windows in the training set corresponding to Far CP data with SNR 3:1. Principal components for the training set appear to have some degree of structured self-similarity in their spectral content, i.e., two or more components have similar subset of spectral components from the training data. Comparing the sequence of CW components present in the first 12 principal components with the parameters in Table 1.1, it follows that the CW components appear in the PCA decomposition grouped loosely in the order of their amplitudes, as expected. Components 13 through at least 20 seem dedicated to target signal, which is present in the ON case and absent in the OFF case.

The corresponding cumulative sums of the ON and OFF eigenvalues are shown in Figure 4.2. The eigenvalues represent the “energy” distribution of the respective training sets among each of the corresponding eigenvectors, where the eigenvectors form a basis for the data. The cumulative energy content for the *m*th eigenvector is the sum of the energy content across all of the eigenvalues from 1 through *m*. This plot gives a visual measure of the variability present in the ON and OFF training sets, where variability is indicated by how many eigenvalues (i.e.,

corresponding eigenvectors) are necessary to account for some amount of the input energy. For example, the first 23 components (ON case), and the first 10 (OFF case) are required to account for 95% of the variance in the respective ON and OFF training sets.

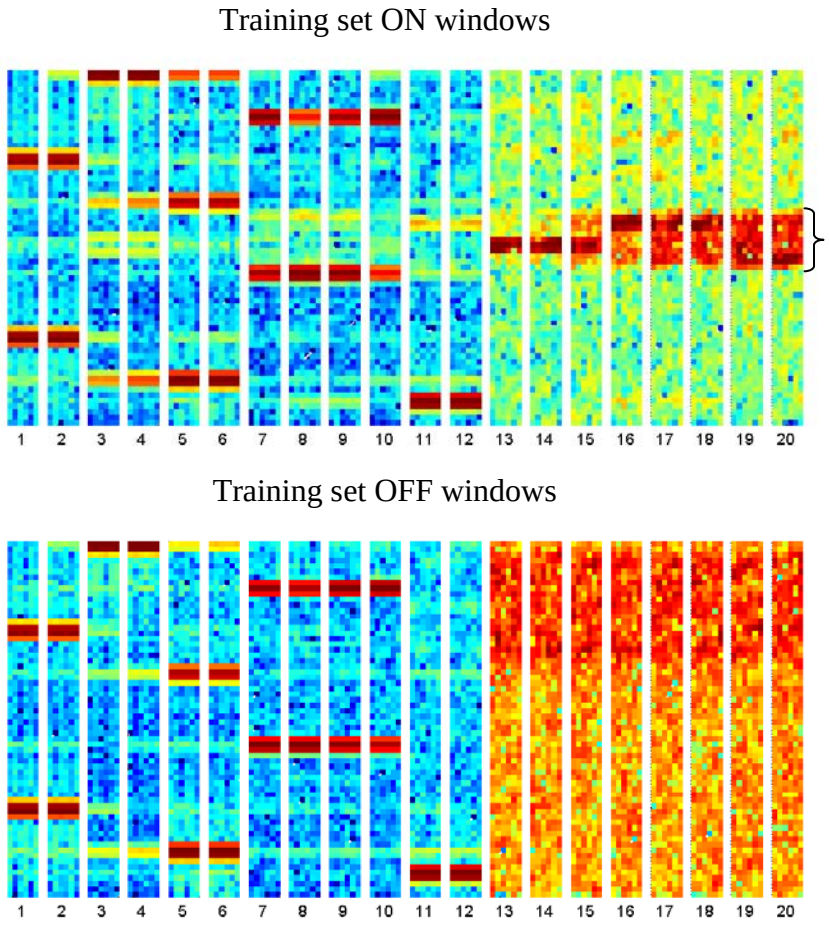


Figure 4.1: Principal components of the training set, arranged in order of decreasing component variance and shown using an identical color map. Shown here are spectrograms (individual vertical strips) of the first largest 20 principal components for the ON (top) and the OFF (bottom) windows in the training set. Every principal component is of length 512 samples. Each spectrogram is constructed with short-time Fourier transform using windows of length 128 with 50% overlap. The corresponding target spectrum is marked with a black brace. Components 13 through at least 20 seem dedicated to target signal, which is present in the ON set and absent in the OFF case.

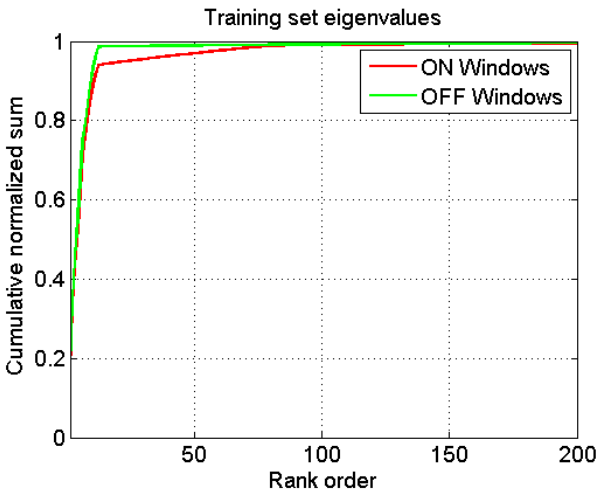


Figure 4.2: Cumulative sum of eigenvalues for the training set (i.e., contribution of each eigenvalue, arranged in decreasing order of magnitude, to the total sum of eigenvalues). Red trace represents ON windows and green trace represents OFF windows.

4.3.1 The data space dimensionality

The completeness of a redundant dictionary is defined as the ratio between the number of dictionary elements and the natural input dimensionality of the data analysis window, 512 samples in this case. A dictionary is *complete* if this ratio is 1:1, *undercomplete* if the ratio is less than 1, and *overcomplete* otherwise.

Choosing a dictionary size that has some degree of overcompleteness with respect to the natural dimensionality, *N*, has shown to be a good choice for learning dictionaries in published work, in particular for reconstruction applications. The performance of such overcomplete dictionaries might, however, be due to the fact that their size was much larger than some notion of "intrinsic" dimensionality of the underlying data space. Intuitively, the dictionary needs to be overcomplete with respect to the dimensionality of the data space as defined earlier, not simply with

respect to the user specified natural dimensionality. In reality, exactly measuring the intrinsic data dimensionality is a very ambiguous problem and a topic of ongoing investigation.

In this thesis, the degree of dictionary overcompleteness is explored from a classification perspective. The fundamental question becomes: does a learned dictionary need to give perfect reconstruction in order to give an accurate classification decision? *This thesis demonstrates that undercomplete learned dictionaries can work for classification, and that they have a number of computational and implementation advantages.* This represents one of the novel outcomes of this thesis project and it will be referenced in subsequent discussion.

4.3.2 Dictionary data space representation fidelity

A *learning convergence* is now defined to quantify the changes of a dictionary in the iterative learning process. Given multiple learning iterations for a particular dictionary, an individual dictionary element, φ_j is said to *converge* if it stops changing significantly from one update, *c*, to the next, as given by:

$$\left\| \varphi_j^c - \varphi_j^{c-1} \right\|_2^2 < \varepsilon, \quad \varphi_j \in \Phi, \tag{4.8}$$

where ε is arbitrarily small. While individual dictionary elements may converge at different rates, the entire dictionary Φ of some size, *K*, can also reach some degree of asymptotic performance after a certain number of learning iterations, where performance is defined from a discriminative point of view. That is, the maximum change across all elements will be bound after some number of learning iterations, *C*,

and the associated classification performance will not vary much. This is further referred to as *dictionary learning convergence*, and can be mathematically expressed:

$$\max_{\varphi_j \in \Phi} \left\| \varphi_j^c - \varphi_j^{c-1} \right\|_2^2 < \beta. \tag{4.9}$$

Measuring learning convergence is straightforward, as it involves direct measurement of Euclidian distances between updates for a given dictionary element, and has been observed in some of the author's publications [16-18].

The measure of convergence in equation (4.9) is appropriate for comparing dictionaries of similar size. The goal in this part of the thesis is to also compare dictionaries of *different sizes*, so learning convergence as defined above is no longer an appropriate metric. The notion of *data space representation fidelity* is more applicable. Given a set of *n* different dictionaries (e.g., different sizes or different random initializations prior to the learning phase), *space representation fidelity* is best when the equivalent *learned* space for all *n* dictionaries closely matches the data space. The hypothesis is that a learned dictionary will define a dictionary space that is closely matched to that of the input data, allowing for good representation and/or classification. That is, the dictionary elements become increasingly more accurate at providing a sparse representation for the data set. The implication is that the learned dictionary space will approximate well the training data space with sufficient training. For a dataset $\{x_p\}_1^N$, let its corresponding residual over some dictionary Φ be given by:

$$r(\Phi, \{x_p\}) = \frac{1}{P} \sum_p \left\| \Phi a_p - x_p \right\|_2^2, \tag{4.10}$$

where $\{a_p\}$ are the optimal sparse solution coefficients for eq. (4.1), and P is the training set size. If the dictionary learning algorithm converges in the data space sense, then:

$$r(\Phi^{c+1}, \{x_p\}) \leq r(\Phi^{c}, \{x_p\}). \tag{4.11}$$

In this work, data space representation fidelity is the underlying decision criterion for the MR classifier, but it is not pursued in the sense of achieving perfect reconstruction. What will be shown is that space representation fidelity is a strong indicator of classification accuracy.

4.4 The learned dictionary space

Recent publications have explored various aspects of dictionary learning, such as sparsity in learning and in reconstruction [80, 111], and the choice of learning algorithms [14, 18, 95, 99, 109, 111]. Less work has been published exploring the choice of learned dictionary size, and it has been relatively common to simply choose a "sufficiently" overcomplete dictionary size with respect to the natural input dimensionality (i.e., the length of data analysis window). In fact, in the closing section of his book [58], Elad includes "setting the proper dictionary redundancy" as a fundamental issue remaining to be studied. This thesis attacks this research problem by hypothesizing a much lower bound for dictionary size and uses experimental results to support it.

Recall that a learned dictionary is not directly generated by an analytical function, but it is rather the result of implicit minimizations on cost or energy functions that are specific to each learning method. A dictionary can be considered a collection of learned matched filters for the training data, but finding quantitative ways to describe and compare dictionaries learned with different methods is not straightforward given the lack of a generating function. The waveforms (or timeseries) of dictionary elements do not visually present any degree of regularity, as shown by inspection in Figure 4.3.

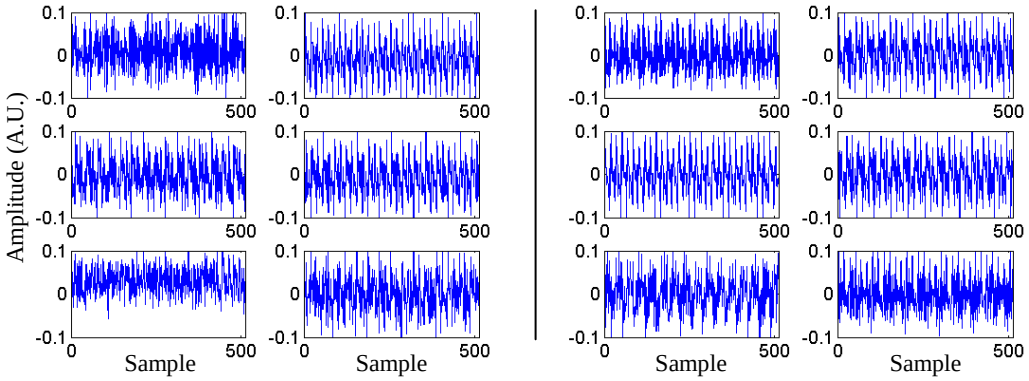


Figure 4.3: Example timeseries of learned elements with N=512 samples from a Hebbian dictionary (left panel) and a K-SVD dictionary (right panel), both with *K*=256 elements.

One way to visually compare dictionaries learned with the two techniques, Hebbian and K-SVD, is to look at their principal components. Spectrograms of the first 20 principal components for a sample Hebbian ON/OFF dictionary pair (Figure 4.4) and a sample K-SVD ON/OFF dictionary pair (Figure 4.5), each with *K*=256 elements, are shown below. These spectrograms give a qualitative view of the learned dictionaries, graphically depicting the "knowledge" present in each of them. For both

Hebbian and K-SVD dictionary types, the first 12-13 principal components are primarily dedicated to the clutter space, after which both Hebbian and K-SVD ON dictionaries exhibit principal components dedicated to the chirped target. In contrast, the principal components for the OFF dictionaries transition from the clutter space directly to the noise space (compare components 14-20 in each case). *A classifier based on the ON/OFF dictionary pair is therefore able to determine the presence or the absence of the target signal in a test window.*

Comparing these PCA spectrograms to those of the training data in Figure 4.1, neither Hebbian nor K-SVD match exactly in terms of spectral content in their respective principal components. However, grouping of only a few CW components at a time is present in the Hebbian case, which is somewhat similar to the training set case. Principal components for the K-SVD case appear more self-similar in their spectral content, while for the Hebbian dictionary they appear less self-similar, and a closer look at the eigenvalues of the dictionary is warranted.

Cumulative sums of the corresponding eigenvalues are shown in Figure 4.6 in order of decreasing magnitude, for each method and each dictionary pair. For the chosen Hebbian pair, the first 104 components (ON case), and the first 85 (OFF case) are required to account for 95% of the variance. In contrast, for the K-SVD pair the first 11 components (ON case), and the first 10 components (OFF case) are needed to capture 95% of the respective dictionary variance.

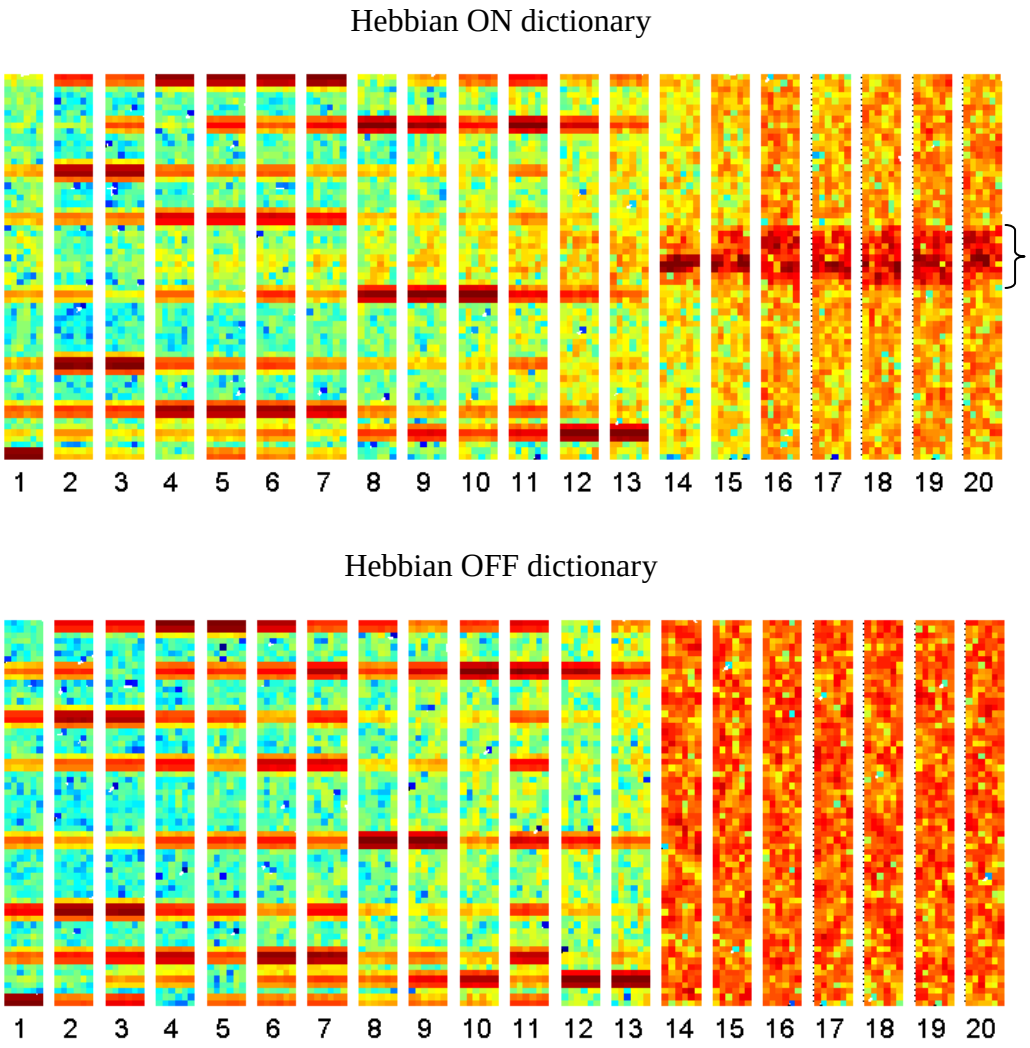


Figure 4.4: Principal components for an example Hebbian ON/OFF dictionary pair with K=256 elements, arranged in order of decreasing component variance and shown using an identical color map. Shown here are spectrograms (individual vertical strips) of the first largest 20 principal components for each of the selected dictionaries. Every principal component is of length 512 samples. Each spectrogram is constructed with short-time Fourier transform using windows of length 128 with 50% overlap. The corresponding target spectrum in the ON dictionary is marked with a black brace.

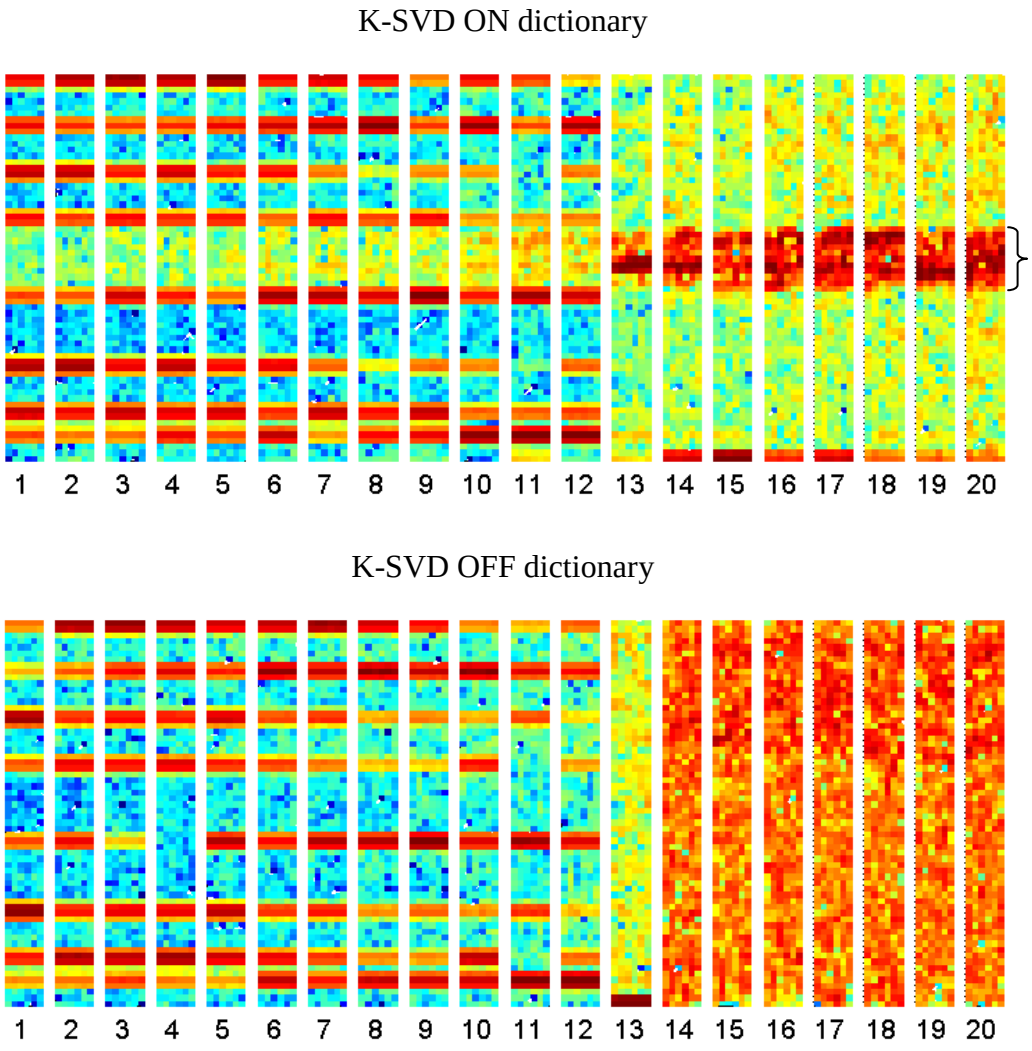


Figure 4.5: Principal components for an example K-SVD ON/OFF dictionary pair with K=256 elements, arranged in order of decreasing component variance and shown using an identical color map. Shown here are spectrograms (individual vertical strips) of the first largest 20 principal components for each of the selected dictionaries. Every principal component is of length 512 samples. Each spectrogram is constructed with short-time Fourier transform using windows of length 128 with 50% overlap. The corresponding target spectrum in the ON dictionary is marked with a black brace.

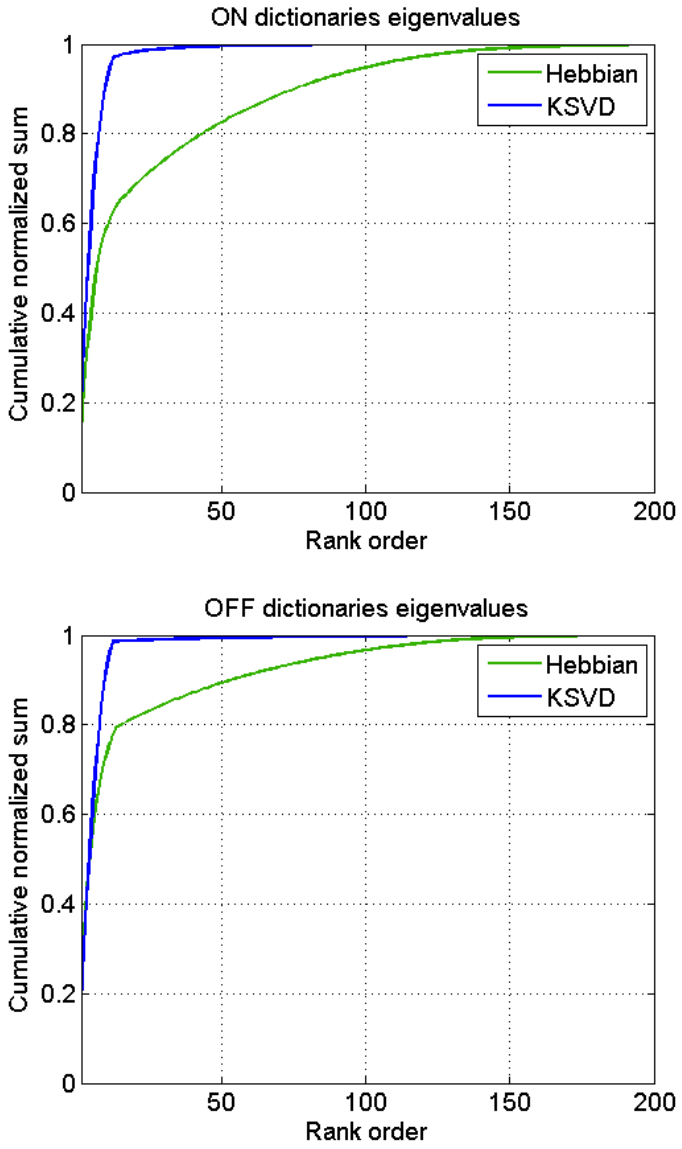


Figure 4.6: Cumulative sum of eigenvalues for ON learned dictionaries (top), and OFF learned dictionaries (bottom) with K=256 elements. Green trace represents Hebbian dictionaries, and blue trace represents K-SVD dictionaries.

The principal component study demonstrates that, for the chosen training data set, there is a higher amount of variability present in a dictionary learned with the Hebbian method compared to one learned using K-SVD. This higher variability could allow for sparser decompositions over the Hebbian dictionary compared to the K-

SVD one for an equivalent approximation fit. It indicates that a Hebbian dictionary might perform better on test data, and subsequent sections confirm this finding.

4.5 Hebbian learning parameters

A number of parameters must be chosen to ensure good performance of the dictionary learning algorithm. For this initial study of the learning parameters, the Far CP data case is again considered first, similar to the parametric dictionary case. The training set size is initially fixed at 1700 ON and 1700 OFF windows, which is a small number and would represent an extreme case of few training data. The question is how much can be learned from such limited data.

The impact of completeness can now be explored for the RF case in greater detail by examining classification performance of dictionaries ranging from undercomplete by a factor of 32 (i.e., 16 elements), to up to twice overcomplete (i.e., 1024 elements), with respect to a natural dimensionality of N=512. Directly related to the choice of dictionary size is another learning parameter: the number of learning iterations C (i.e., how much training does the dictionary need). Lastly, the sparsity factor of the approximation, L, (see definition in Section 4.1) can also impact the quality of the learning and of the classification, as it directly controls how much "information" or detail is extracted from the data.

These three parameters are discussed below using classification accuracy as performance metric. In reporting accuracy, false positives and false negatives are equally weighted. For applications in which false positives and false negatives are not

assigned equal weight, the data from which the dictionary is learned could be chosen to minimize either false positive rate or false negative rate.

4.5.1 Number of learning iterations, *C*

It is generally accepted that K-SVD learning converges relatively fast in terms of representation accuracy [10, 95], and usually in fewer than 30 iterations [10]. Similar convergence behavior was observed by the author when using K-SVD dictionaries to classify other RF simulated data [16]. To similarly study the learning convergence in classification for Hebbian dictionaries, the dictionary size and learning sparsity factor are fixed and a wide range of learning iterations is considered (Figure 4.7).

Two sizes are initially set at an undercomplete value of K=256 (top panel) and a complete value of K=512 (bottom panel), and the learning sparsity factor (i.e., the sparsity factor used during the learning process) in both cases is L_{learn}=45. This value for L_{learn} was chosen based on the results in [16] as a sparsity factor value that is large enough to not impact accuracy performance, thus allowing a more straightforward study of the effect of learning iterations. Values of C between 25 (i.e., around the number of iterations needed for K-SVD to converge) and up to 600 are used in each case, and 10 ON/OFF dictionary pairs with different random seeds are learned for each value of C. The boxplots in Figure 4.7 summarize the accuracy of these dictionaries for classification when tested against unseen Far CP SNR 3:1 test data, where the variance and median accuracy at every value of C are calculated over the

respective 10 dictionary pairs. In the top panel, for K=256, the accuracy improvement levels off once the number of learning iterations C is roughly the same as the number of dictionary elements K to be learned (marked by the vertical line). The maximum reached classification accuracy is ~0.998 and first occurs at learning iteration 250. The overall accuracy variance (i.e., height of the boxplot) decreases between 75 and 350 learning iterations, and begins increasing again past 400 iteration, which would be a sign of overfitting (i.e., overlearning) the training data.

In the bottom panel, for K=512, the accuracy improves more slowly and to a slightly smaller maximum value, and the variance is larger up to C=500 learning iterations, as one would expect with the greater degrees of freedom associated with a larger dictionary. Clearly, for both dictionary sizes, Hebbian learning requires much more than ~30 iterations to converge (compared to K-SVD learning) for this fixed training set size. In the K=512 case, it is also apparent that the performance worsens as the learning iterations increase by a larger extent than in the K=256 case, an indication that more training data is needed for the larger dictionary. Therefore, even from limited training data one can learn a relatively good small dictionary using many learning iterations (e.g., C=250 for K=256), but as K increases it may no longer be possible to do so.

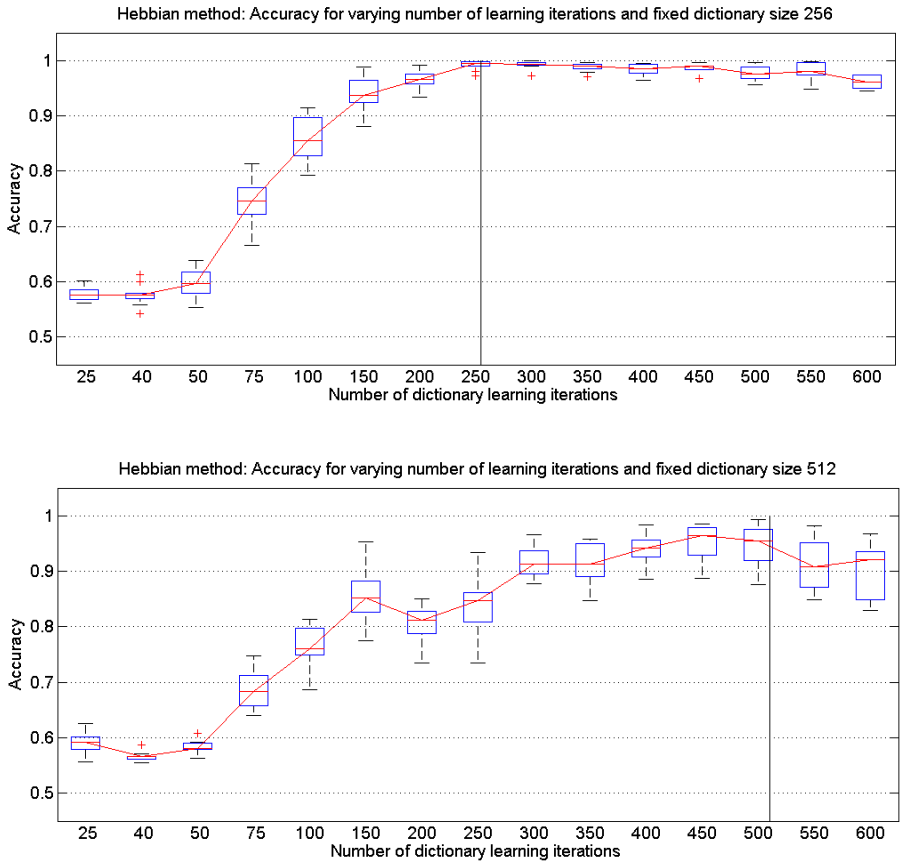


Figure 4.7: Classification accuracy for Hebbian dictionaries of size K=256 elements (top) and K=512 elements (bottom) for different numbers of learning iterations C, where each iteration considers all P=3400 ON and OFF training vectors. The 10 ON/OFF dictionary pairs in each boxplot are learned on high amplitude target training data with sparsity factor L=45 and used to classify unseen high amplitude test data.

4.5.2 Dictionary size, *K*

Now the performance of the two methods is compared as the dictionary size K changes. For the K-SVD case, 10 ON/OFF dictionary pairs with sizes of K between 16 and 1024 elements are learned from Far CP SNR 3:1 training data, with C=25 learning iterations. Figure 4.8 shows the accuracy of the resulting dictionaries in classifying unseen high target amplitude (SNR 3:1) test data. The learning behavior is

as follows: classification accuracy improves gradually as the size of the dictionary increases (near K=128) and then remains relatively similar for a wide range of K. An important observation is that a K-SVD learned dictionary that is undercomplete with respect to the input dimensionality (e.g., dictionary size of K=192 vs. input dimensionality of 512) can achieve accuracy comparable to complete or overcomplete K-SVD dictionaries trained on the same amount of training data.

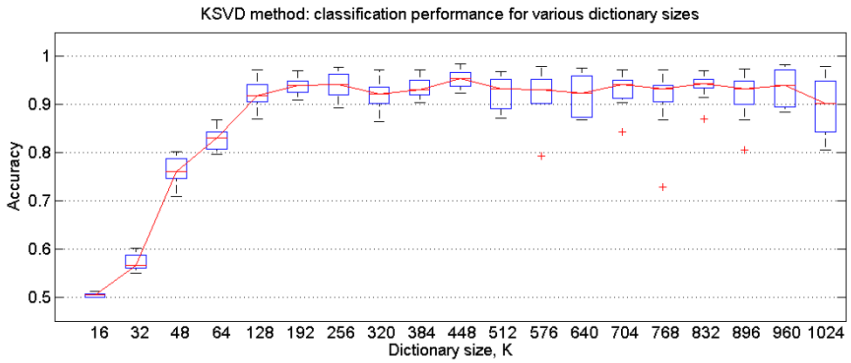


Figure 4.8: Classification accuracy for different dictionary sizes K. The boxplots summarize 10 ON/OFF dictionary pairs learned with the K-SVD method using a constant number of learning iterations C=25. The training and test data contained a high amplitude target and the sparsity factor was L=45.

For the Hebbian case, the required amount of dictionary training (i.e., minimal number of learning iterations) increases proportional with the number of dictionary elements, K, for a fixed training set size (based on Figure 4.7). An illustration of the impact of insufficient training on the dictionary is shown in Figure 4.9. The dictionary pairs shown here were all learned with the Hebbian method using a constant number of learning iterations C=250 and a sparsity factor of L_{train}=45.

Median performance degrades considerably as the dictionary size increases past K=256, and the variance across the 10 pair set also becomes larger.

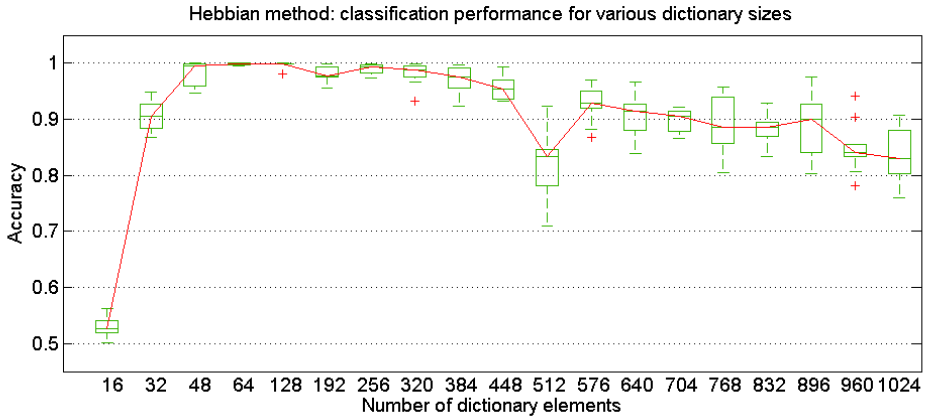


Figure 4.9: Classification accuracy for different dictionary sizes K. The boxplots each summarize 10 ON/OFF dictionary pairs learned with the Hebbian method using a constant number of learning iterations C=250. In both plots the training and test data contained a high amplitude target and the sparsity factor was L=45.

Figure 4.9 suggests that an undercomplete Hebbian learned dictionary may perform as well as a complete or overcomplete Hebbian dictionary, given an optimal amount of training, since the improvement in classification peaks for values of K>64 (for C=250). However, the classification performance obtained so far with sufficiently trained undercomplete dictionaries is a compelling reason to focus on undercomplete (K<512) and complete (K=512) Hebbian dictionaries.

Figure 4.10 now shows the classification accuracy for these dictionaries on Far CP SNR 3:1 test data, where each set of 10 dictionary pairs is computed with values of C appropriate for the respective dictionary size. For a range of dictionary sizes $K \leq 512$ and given sufficient training, Hebbian dictionaries out-perform K-SVD dictionaries on this data set. Additionally, the asymptotic classification performance

for the Hebbian case is reached more rapidly than in the K-SVD case, i.e., for a smaller dictionary. These Hebbian undercomplete dictionaries also exhibit a better learning convergence variance across the 10 dictionary pairs compared to the K-SVD case, e.g., K={64, 128}.

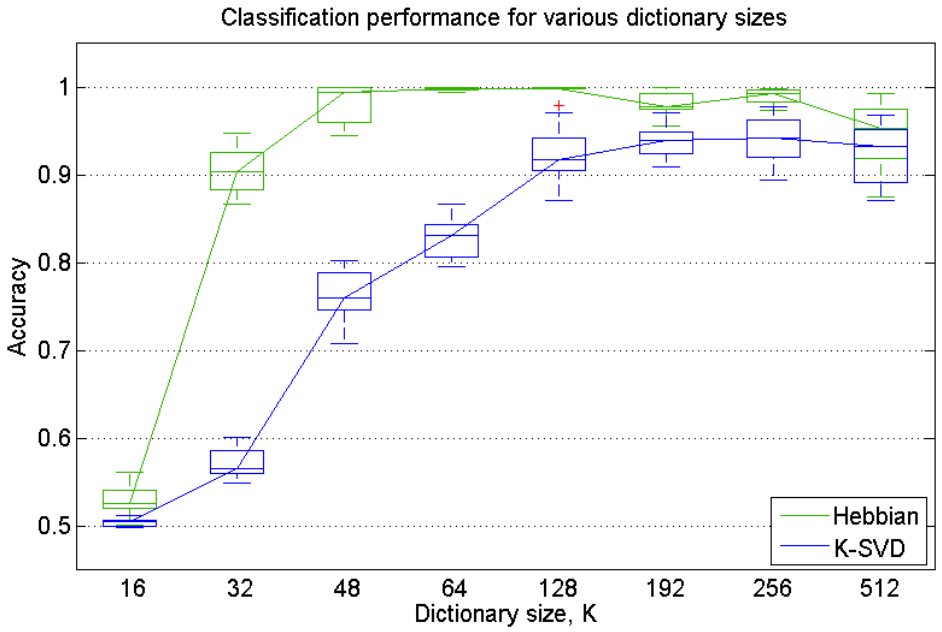


Figure 4.10: Classification accuracy for different dictionary sizes K. The boxplots each summarize 10 ON/OFF dictionary pairs learned with the Hebbian method (green boxplots) using C=250 for K=16:256 and C=500 for K=512; and with the K-SVD method (blue boxplots). Training and testing is done using Far CP SNR 3:1 data, with sparsity factor L=45.

For both dictionary learning approaches, these results suggest that classification based on a learned dictionary may succeed with dictionaries that are undercomplete with respect to the natural dimensionality of the training vectors.

4.5.3 Learning sparsity factor, L_{train}

The ability of the adaptive, data-learned dictionaries in capturing the signal with sparse, efficient features is illustrated in Figure 4.11. Equal numbers of ON and

OFF windows were selected from among SNR 3:1 Far CP test data, and matching pursuit was used to construct two 50-element representations of each window: one with an ON 256-element dictionary (C=250 learning iterations), the other with the respective OFF dictionary. Figure 4.11 shows semilog plots of residual energy at each matching pursuit iteration (i.e., the residual energy as dictionary elements are added to the representation), averaged over all the ON windows (left panel) and over all the OFF windows (right panel). For comparison purposes, also shown is the residual energy for the ON and OFF cases using matching pursuit in an overcomplete chirped Gabor dictionary.

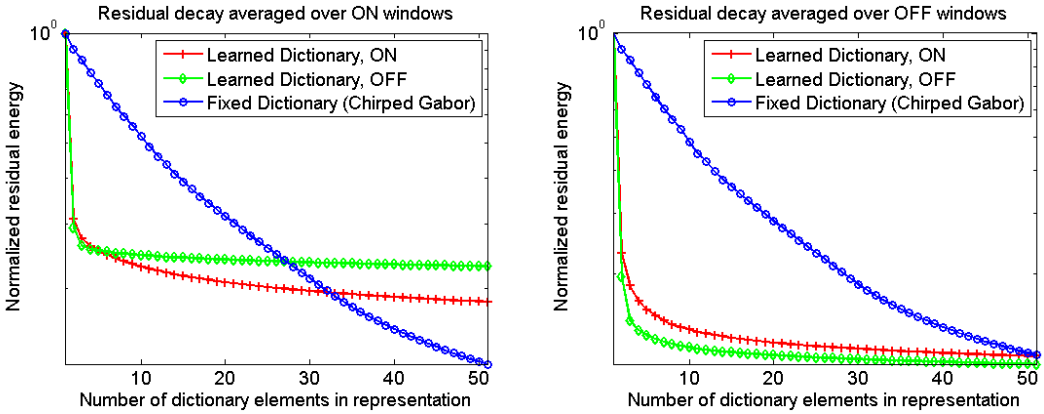


Figure 4.11: Residual decay in decomposition over a Hebbian dictionary pair learned from SNR 3:1 Far CP data versus an overcomplete Gabor dictionary for ON window (left), and OFF window (right).

For the case of ON data windows, the residual from the ON dictionary decomposition (red line), is consistently smaller than the residual over the OFF dictionary (green line) after the first few iterations. Both learned dictionary residuals

decay more rapidly that the equivalent residuals over the Gabor dictionary (blue line), but do not achieve reconstruction errors that are as good. This is evidenced by the fact that residual energy associated with the Gabor dictionary drops beneath that of the learned dictionaries as the number of elements approach 50. The poorer reconstruction performance is intuitive given that the particular Gabor dictionary has more than 10 times the number of elements compared to the 256-element learned dictionaries. The right-hand plot shows the same set of averaged residuals, this time calculated for OFF data windows. The minimum residual classifier can correctly choose the window label in each case without ambiguity after the first few matching pursuit iterations, as the corresponding dictionary residual is always lowest.

One observation in Figure 4.11 is that the sparsity factor (number of features) used in signal classification, $L_{class,}$ can be quite different than the sparsity factor used to learn the dictionary, L_{train}, and in particular it can be smaller, to obtain a good classification. This will be discussed again in subsequent sections. Furthermore, a study was done [16] assessing the impact of L_{train} on the classification performance for K-SVD dictionaries and found that the classification performance of the dictionary is insensitive to the value of L_{train} once the sparsity factor is greater than some lower bound. The learning sparsity factor for the Hebbian case is similarly explored in Figure 4.12, which shows classification performance for a selected dictionary size as the value of L_{train} is increased from 2-sparse (i.e., very coarse approximations) to 60-sparse approximations (i.e., finer approximations).

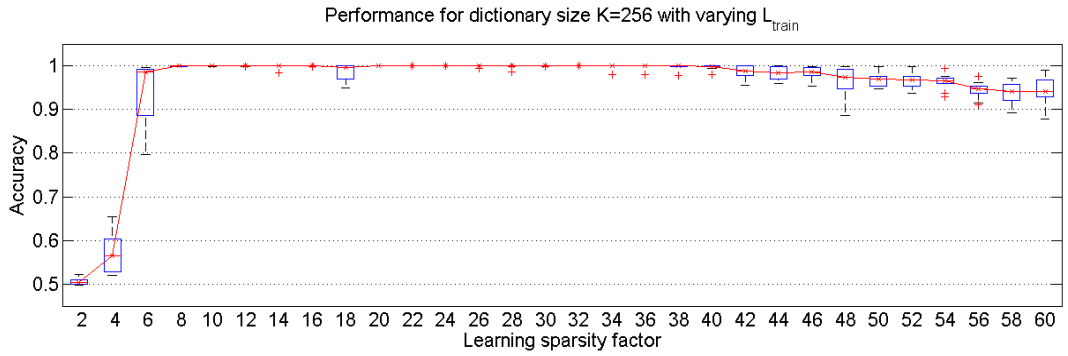


Figure 4.12: Classification accuracy of size K=256 Hebbian dictionary as the learning sparsity factor, L_{train}, increases from 2 to 60. The dictionaries are learned on SNR 3:1 Far CP training data and used to classify SNR 3:1 Far CP data using L_{class} equal to the respective L_{train}.

In Figure 4.12, sets of 10 dictionary pairs were learned with increasing sparsity factors L_{train} while keeping the number of elements fixed at K=256 and the amount of training at C=250 learning iterations. Every dictionary set is applied to test data with a classification sparsity factor L_{class} equal to the respective $L_{train.}$ Each boxplot represents accuracy statistics across a set of 10 dictionary pairs. The red line connects the median accuracy values for each set.

The plot suggests that the performance of the dictionary has a region of insensitivity to the value of L_{train} once the learning sparsity factor is greater than some lower bound. This is important when the data is unfamiliar and we wish to make optimal choices for dictionary learning parameters. Similar to the K-SVD case [16], in Figure 4.12, once L_{train} is greater than ~8 the accuracy remains within the same interval until L_{train} ~40, with its median values hovering around 0.99. In other words, the same classification performance is obtained with coarser approximations (i.e., smaller L_{train}) as with finer approximations, but without the added computational cost.

This supports the hypothesis that perfect reconstruction is not necessary for perfect classification.

Section 4.5 focused on studying the impact the various learning parameters has on the classification accuracy, for both Hebbian and K-SVD methods. This study has revealed what appears to be an opportunity to leverage the respective strengths of the different types of dictionaries by forming a so-called *hybrid dictionary*.

4.6 Hybrid learned dictionaries

Figure 4.10, comparing classification performance for the undercomplete and complete K-SVD and Hebbian dictionaries, showed that the optimally trained Hebbian dictionaries outperformed the corresponding K-SVD ones, but do so at an increased computational cost due to the high number of learning iterations required. K-SVD dictionaries learned with C=25 iterations and sizes K=256 and K=512 have a median classification accuracy of 0.94, but consistently exhibit a wide variation across the 10 differently seeded pairs of dictionaries (Figure 4.8). For similarly sized Hebbian dictionaries learned with the same number of C=25 iterations (Figure 4.9), the classification accuracy is only slightly better than chance.

A hybrid dictionary method is now proposed in an attempt to combine the strengths of the two methods and compensate for their respective weaknesses. Both methods require some prior initialization of the dictionary, usually random dictionary elements or imprinting of training data vectors. The goal of the hybrid method is to improve the prior knowledge for a Hebbian dictionary in order to accelerate the

learning convergence of the algorithm. *The specific novel solution proposed in this section is to use an intermediate K-SVD learned dictionary that has not fully converged to seed the Hebbian dictionary.* Two intermediate K-SVD stages are considered, one after 1 learning iteration (1 K-SVD seed), and one after 3 learning iterations (3 K-SVD seeds). These values are chosen to rapidly determine whether the hybrid dictionary would be useful. The main intuitive advantage of a hybrid dictionary is that Hebbian leaning must only fine-tune dictionary elements obtained via K-SVD, instead of learning the entire dictionary from a random initialization.

Figure 4.13 augments the results in Figure 4.7 with the accuracy performance obtained by hybrid dictionaries on the Far CP data as a function of the number of learning iterations. The top two panels of Figure 4.13 show accuracy for the K=256 case with 1 and 3 K-SVD seeds, and the lower two panels similarly show accuracy for the K=512 case with 1 and 3 K-SVD seeds. For ease of visualization, the green trace represents the median accuracy obtained across the 10 simple Hebbian dictionaries of Figure 4.7 at the same number of learning iterations.

First off, the variance across the 10 pair sets is reduced compared to the equivalent Figure 4.7 panels, in particular at lower number of learning iterations. The asymptotic behavior is reached much faster (i.e., by 50-75 learning iterations, compared to 250), and in the K=256 case appears more stable compared to the K=512 case. The median accuracy performance at C=25 learning iterations jumped from ~0.58 in the simple Hebbian case to ~0.97 in the hybrid Hebbian case seeded with a 3-iteration K-SVD dictionary.

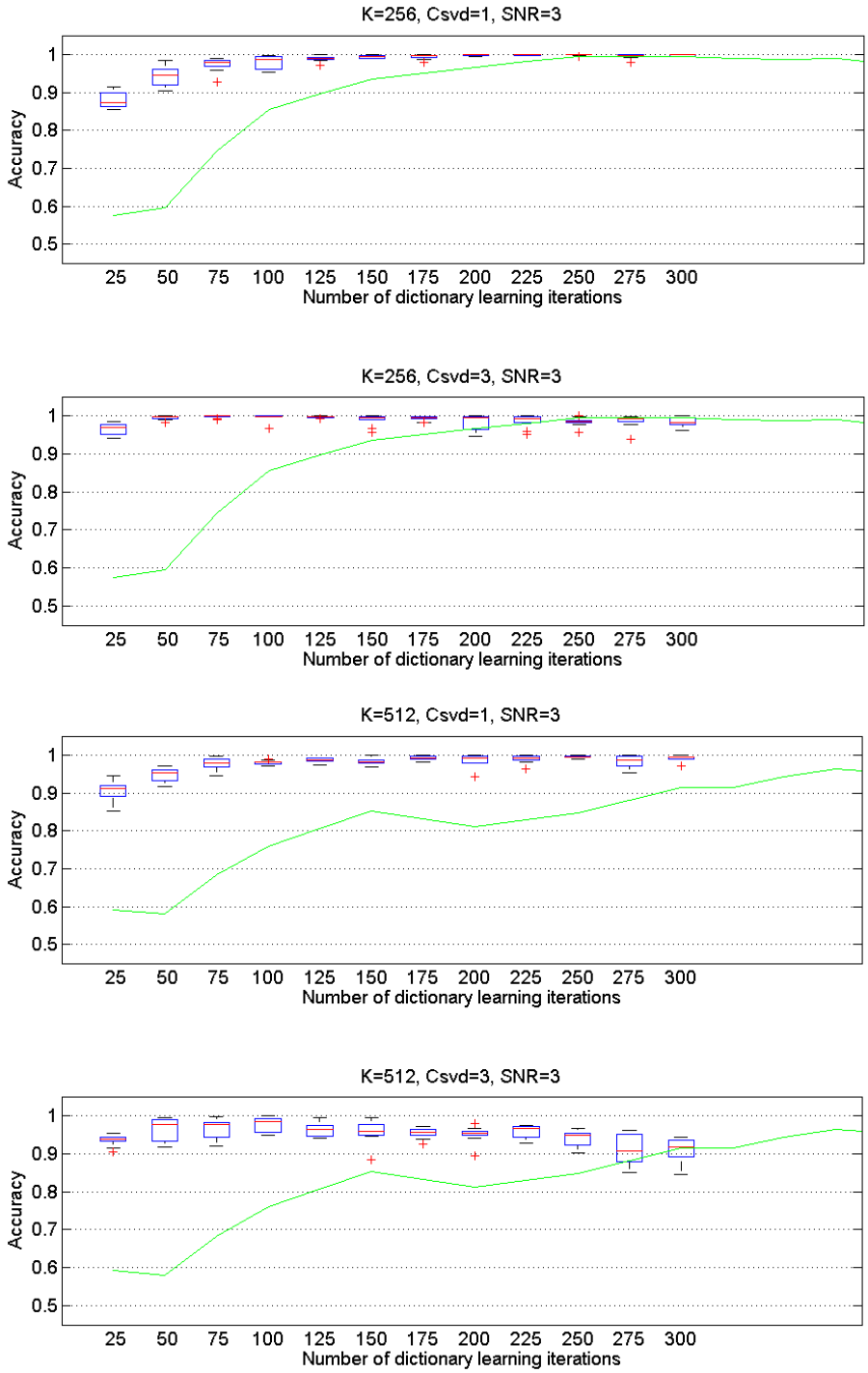
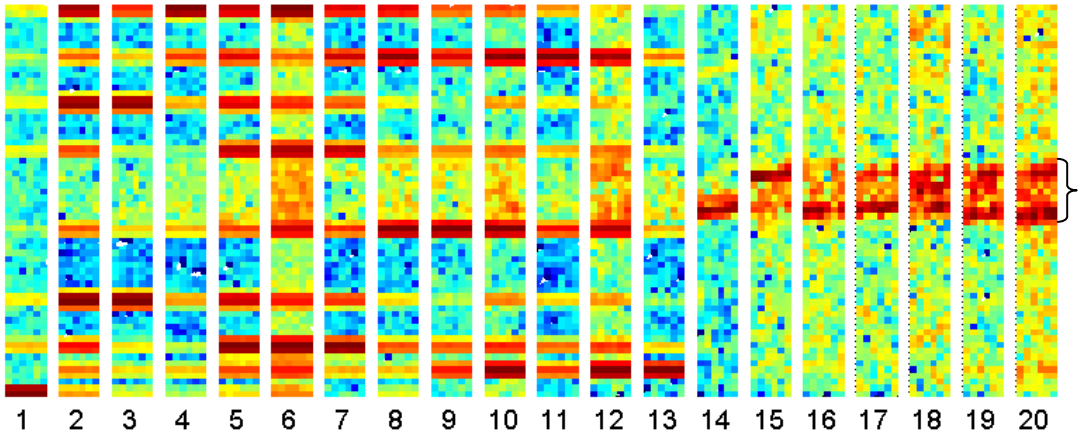


Figure 4.13: Boxplots each summarize accuracy performance for 10 ON/OFF hybrid pairs. Green traces represent median accuracy from Figure 4.7 for simple Hebbian dictionaries. The hybrid dictionaries reach asymptotic classification behavior much faster, as anticipated.

An interesting question is whether these hybrid dictionaries are more K-SVD-like or more Hebbian-like in terms of their structure. Spectrograms of the first 20 principal components for hybrid ON dictionaries with K=256 elements are shown in Figures 4.14-4.15 on a fixed color scale. Figure 4.14 shows the ON dictionary with 1 K-SVD seed at 25 learning iterations (top) and at 250 learning iterations (bottom). Similarly, Figure 4.15 shows the ON dictionary with 3 K-SVD seeds at 25 learning iterations (top) and at 250 learning iterations (bottom). All spectrograms use the same fixed color scale as the spectrograms of Figures 4.4-4.5 for easy comparison.

At C=25 iterations, there is more K-SVD-like structure in the hybrid dictionary with 3 seeds, as expected (Figure 4.15 top panel) compared to the hybrid one with 1 seed (Figure 4.14 top panel). The hybrid dictionaries morph more into Hebbian dictionaries as the number of learning iterations increases (bottom panels of Figures 4.14-4.15). The cumulative sum of the ordered eigenvalues for the hybrid dictionaries (Figure 4.16) also exhibits similar variability to that shown in Figure 4.6. That is, the number of principal components needed to capture 95% of the variability in the dictionary is higher, especially in the ON case, and similar to the simple Hebbian case. Recall that for K-SVD dictionaries only ~11 principal components are needed to capture 95% of the dictionary variance.

Hybrid ON dictionary: 1 K-SVD seed; *C*=25



Hybrid ON dictionary: 1 K-SVD seed; *C*=250

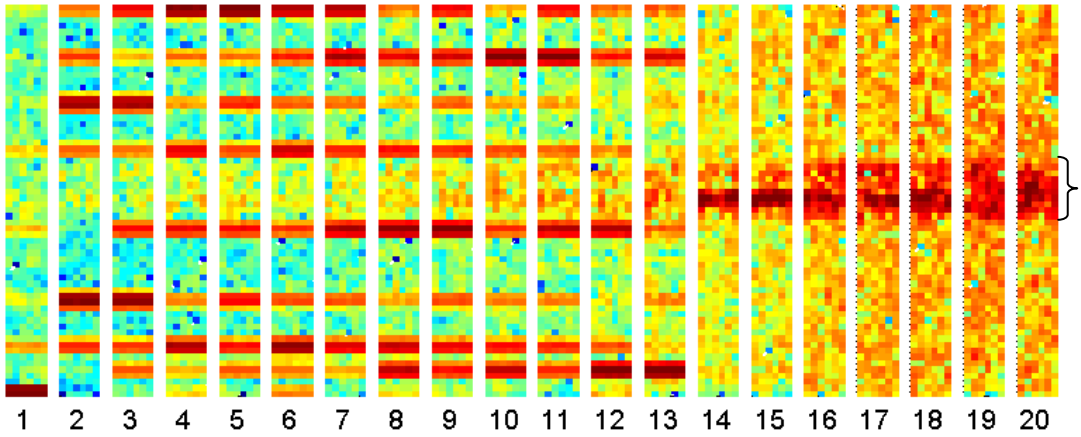


Figure 4.14: Principal components for example hybrid ON dictionary with *K*=256 elements and 1 K-SVD seed at 25 learning iterations (top) and at 250 learning iterations (bottom). Spectrograms (individual vertical strips) are comprised of the first largest 20 principal components for each of the selected dictionaries. Every principal component is of length 512 samples. Each spectrogram is constructed with short-time Fourier transform using windows of length 128 with 50% overlap. Target spectrum is marked with black brace.

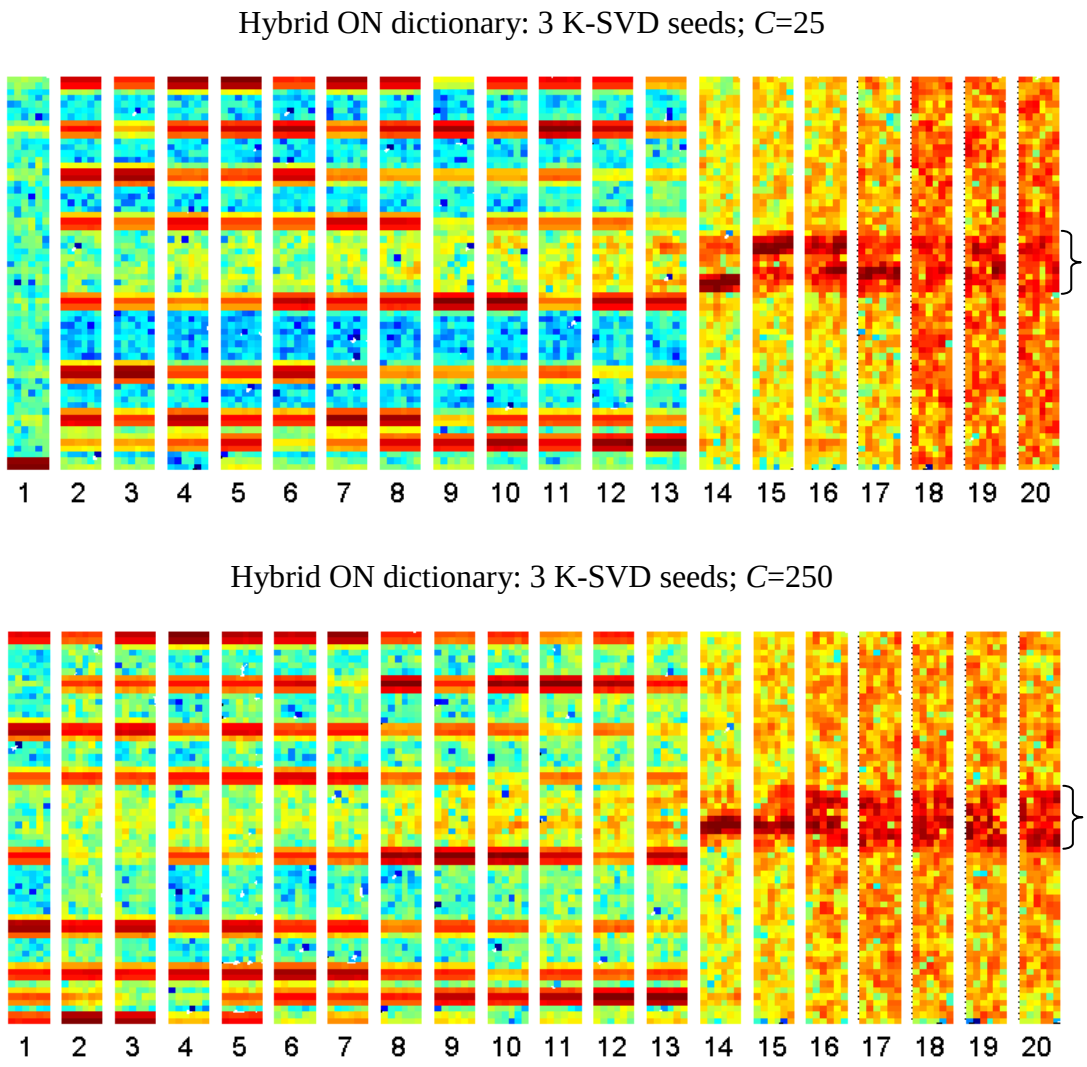


Figure 4.15: Principal components for example hybrid ON dictionary with K=256 elements and 3 K-SVD seeds at 25 learning iterations (top) and at 250 learning iterations (bottom). Spectrograms (individual vertical strips) are comprised of the first largest 20 principal components for each of the selected dictionaries. Every principal component is of length 512 samples. Each spectrogram is constructed with short-time Fourier transform using windows of length 128 with 50% overlap. Target spectrum is marked with black brace.

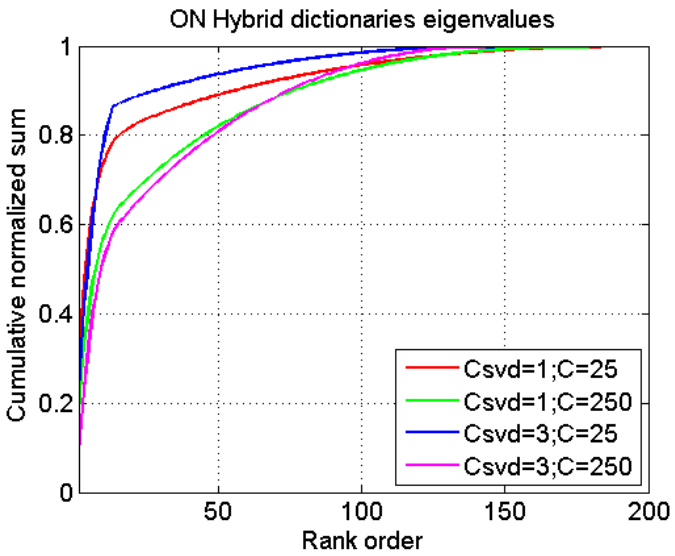


Figure 4.16: Cumulative sum of eigenvalues for ON hybrid dictionaries with *K*=256 elements with 1 (red and green traces) or 3 (blue and magenta traces) K-SVD seeds, after 25 Hebbian learning iterations (red and blue traces), and after 250 Hebbian learning iterations (green and magenta traces).

The hybrid dictionaries are more Hebbian in nature than K-SVD, retain the same high variability that enables higher performance for the Hebbian case, and require a much lower number of minimum learning iterations to reach asymptotic classification performance. They also appear to be more stable for a wider range of learning iterations (i.e., exhibit flat accuracy), which is a result that will prove useful in classification of noisier data in Chapter 5.

4.7 Minimum Residual Classifier (MRC)

Learned dictionaries can be used in a straightforward fashion for classification with a minimum residual classifier (MRC) [15, 16]. This section takes a closer look at the classifier and its robustness with respect to feature selection and SNR conditions. The MRC is a simple, linear classifier, and the classification features it uses are

embedded in the learned dictionary. In this thesis, the training data contains only two classes, leading to a pair of dictionaries: one ON dictionary, one OFF dictionary, but extensions could be made to multiple classes. The MRC assigns to each test input the label corresponding to the dictionary yielding the smallest matching pursuit residual energy. Following the learning process and its different parameters to be optimized, the next question to consider is the fidelity of the classification decision with the specific classification scheme chosen (i.e., the MRC). The performance of the classifier is evaluated now in terms of its response to the number of features selected in testing (i.e., L_{class}), its confidence (i.e., how "sure" is the decision), and its robustness to variation in discrimination threshold (i.e., "receiver operating characteristic"). All three are evaluated in turn.

4.7.1 Classification sparsity factor, L_{class}

The sparsity factor (maximum number of features) used in signal classification, L_{class}, can be different than the L_{train} used to learn the dictionary, as previously noted in Section 4.5.3. In the case shown in Figure 4.11, the 256-element dictionaries were learned with fixed sparsity factor L_{train} =45. It is clear from Figure 4.11, however, that a correct classification decision can be produced with L_{class} values as low as ~12. Thus, building a dictionary with significantly more elements than what is needed for accurate classification can be wasteful.

Figure 4.17 is a quantitative view of the classification process for SNR 3:1 Far CP test windows. Shown here are the residual differences seen by the minimum

residual classifier for a sample pair of ON/OFF dictionaries for both Hebbian (left side) and K-SVD (right side) dictionary methods, as a function of the sparsity factor used in classification. The ON-minus-OFF difference of residual energy is calculated every time a dictionary element is added to the sparse representation of the test data (i.e., classification sparsity factor, L_{class}, increases). This residual energy difference therefore provides the classification decision at every greedy matching pursuit iteration, or as the value of L_{class} increases. The minimum residual classifier assigns the correct label when the ON-minus-OFF residual energy difference is negative for ON windows, and positive for OFF windows. For ON windows (top panel) this residual difference stays well below zero after just a few matching pursuit iterations for both dictionary learning methods. For OFF windows (lower panel), the magnitude of the residual difference is smaller and, for the K-SVD case, not always positive. The classification accuracy corresponding to the dictionaries shown in Figure 4.17 using residuals at L_{class}=45 is 0.96 for the K-SVD dictionary and 0.99 for the Hebbian dictionary.

The idea that the classification sparsity factor can be different and, in particular, smaller than the learning sparsity factor is supported by Figure 4.17, especially for the Hebbian case (left panels). This dictionary pair was learned using L_{train} =45, but if classification is made using, for example, a classification sparsity factor L_{class} =25, the performance (accuracy=0.998, false positive rate=0.003, false negative rate=0) is equivalent to that of L_{class} =45 (accuracy=0.993, false positive rate=0.01, false negative rate=0.01), and without the added computational cost.

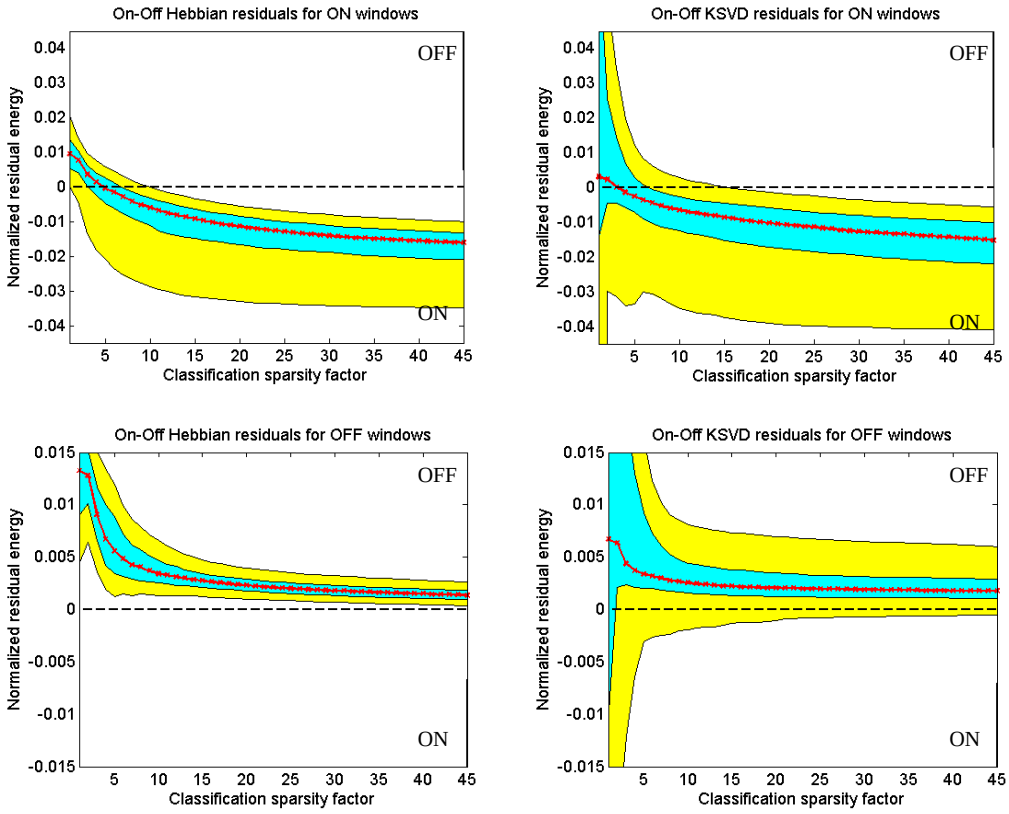


Figure 4.17: Test timeseries residual differences between ON and OFF dictionaries with K=256 elements seen by the MR classifier at each matching pursuit iteration (i.e., as the value of L_{class} increases on the x-axis). An example ON/OFF dictionary pair is selected to illustrate each learning method, and shown are residuals for fully ON (top panel) and fully OFF (bottom panel) test windows. For every additional dictionary element included in the sparse decomposition, the median residual difference over the test windows (red cross line) is estimated, including the 25^{th}–75^{th} percentiles (blue area) and the 5^{th}–95^{th} percentiles (yellow area). The left panels show residual differences for the selected Hebbian dictionary pair, while the right panels show the residual differences for the selected K-SVD dictionary pair. The Hebbian ON/OFF dictionary pair is learned with C=250 learning iterations, sparsity factor L_{train} =45, and trained and tested on Far CP data with SNR=3:1. The K-SVD ON/OFF dictionary pair is learned with C=25 learning iterations, sparsity factor L_{train} =45, trained and tested on Far CP data with SNR=3:1. The region above the dashed line at y=0 corresponds to assigning a label of OFF, the region below to assigning a label of ON.

4.7.2 Residual decision maps

Classifier performance under changing noise conditions can be visualized in the manner of Figure 4.18. Residual decision maps plot the residual energy of one dictionary (OFF on the *y*-axis) against the other (ON on the *x*-axis), giving a graphical illustration of the classification process (i.e., choosing the dictionary yielding higher data representation fidelity, which results in the smaller residual). The decision boundary is marked by the diagonal dashed line (i.e., where both ON and OFF residuals are equal, meaning both dictionaries are reaching similar data representation fidelity). Shown here are minimum residual classifier decision maps for SNR 3:1 Far CP test data (top left), SNR 1:1 test data (top right), and SNR 0.3:1 test data (bottom), aggregated over all test data windows. A single Hebbian dictionary pair with K=256 elements is used in Figure 4.18, and is specifically selected among the 10 pairs as giving maximum accuracy on SNR 0.3:1 test data. For each window, the classifier decision is plotted at coordinates given by the pair of residual energies with respect to the ON and OFF dictionaries, and color coded according to its correctness. This results in two true class clusters: one for ON windows (green), and one for the OFF windows (blue). Misclassified windows are marked in red (False ONs) and magenta (False OFFs). Ideally, a classifier would provide good, clear separation of the clusters, with little or no possibility of misclassification.

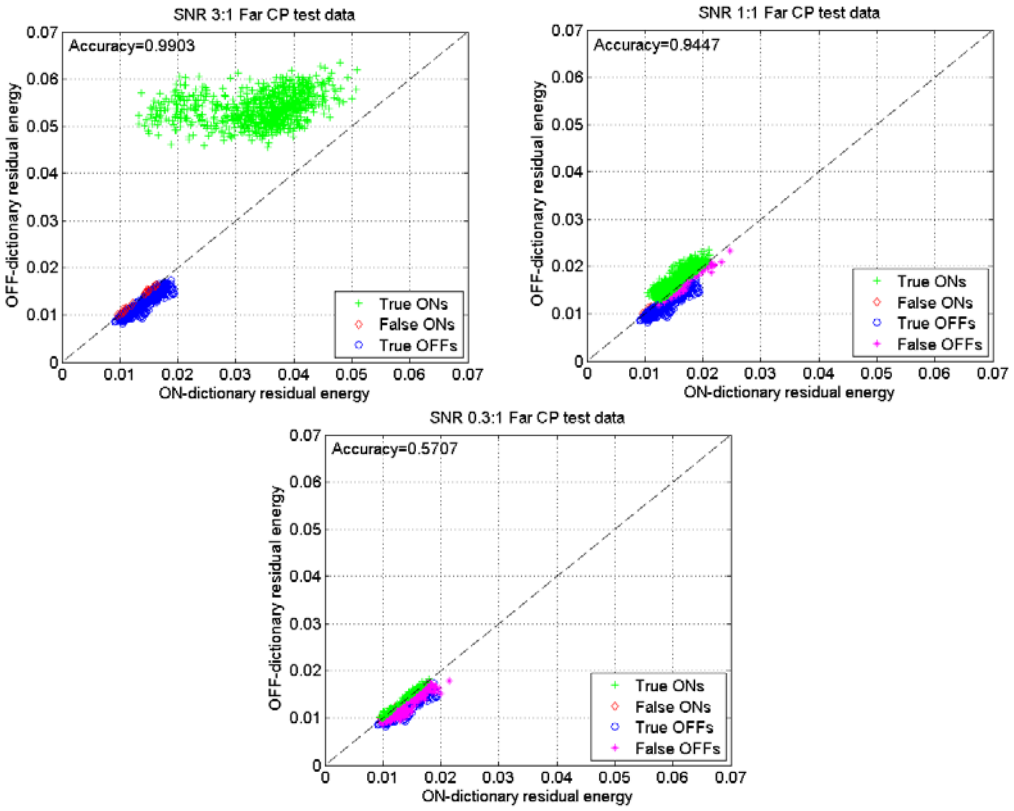


Figure 4.18: Classification maps in the (ON, OFF) residual plane for a selected Hebbian dictionary of size K=256 applied to data windows of SNR 3:1 data (top left), SNR 1:1 data (top right), and SNR 0.3:1 data (bottom). Decision boundary rests along the 45^{o} diagonal.

The decision maps in Figure 4.18 show the minimum residual classifier provides class separation with a good margin of confidence (i.e., wide separation in the residual plane between the clusters of ON and OFF windows) in the SNR 3:1 case, and less confidence in the SNR 1:1 and 0.3: 1 cases. The specific classification accuracy of the dictionary pair used in Figure 4.18 was 0.9903 for SNR 3:1 data, 0.9447 for SNR 1:1 data, and 0.5707 for SNR 0.3:1 data. In the SNR 3:1 case, there is strong decision confidence, resulting in high accuracy and no false OFFs (i.e., no misses) for the selected dictionary. For SNR 1:1 test data, even though the accuracy is

still high, the clusters are actually touching, and both ON and OFF clusters rest against the diagonal decision boundary. In the SNR 0.3:1 case the accuracy is only slightly better than chance and the cluster separation is quite poor.

4.7.3 ROC curves

A formal way of assessing the classifier's performance is to look at its equivalent receiver operating characteristic (ROC) curve (Figure 4.19). The ROC curve is a graphical plot of the true positive rate (i.e., sensitivity), vs. false positive rate for a binary classifier as its discrimination threshold is varied. Evaluation is made with respect to the main diagonal dividing the ROC plane. Points above the diagonal represent good classification results (better than random), points below the line poor results (worse than random). The more each curve tends toward the left and top edges of the plot, the better the classification. Perfect classification is achieved in the top left corner of the ROC plane.

The classification decision of the MRC can be mathematically expressed as the Heaviside step function of the ON/OFF residuals $H(res_{OFF} - res_{ON} + \theta)$, where

$$H(x) = \int_{-\infty}^{x} \delta(t)dt. \tag{4.12}$$

Given that the maximum values for either residual is 1, the discrimination threshold θ is varied between ±1 with a moderate resolution step between [-1, -0.2) and (0.2,1], and a fine resolution step of 0.005 in the [-0.2, 0.2] interval. Figure 4.19 shows the plot of true positive rate vs. false positive rate (1 − true negative rate) for

the MRC, as Θ changes, for a Hebbian dictionary (top left) and a K-SVD dictionary (top right). Both example dictionaries are of size K=256, and a range of classification sparsity factors, L_{class}={25,30,35,40,45}, is considered for each dictionary to classify Far CP SNR 3:1 test data. A zoom of the Hebbian ROC curve is also shown (bottom left) to capture the variability around the top right corner, and similarly for the K-SVD case (bottom right).

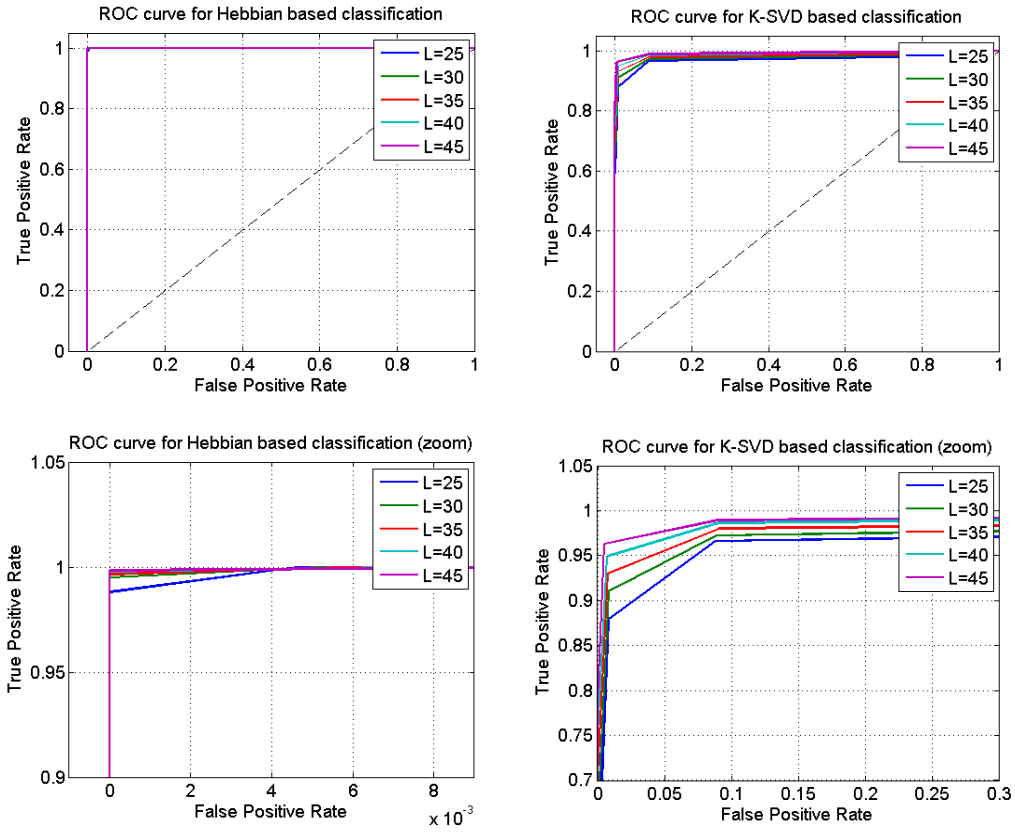


Figure 4.19: MR classifier ROC curves for a Hebbian dictionary with K=256 and C=250 (top left) and a K-SVD dictionary K=256 (top right). Zooms of respective ROC plots are shown in the bottom panel. Both dictionaries are learned with L_{train} =45. Classification sparsity factors of L_{class}={25,30,35,40,45} and considered for each dictionary.

The plots in Figures 4.17-4.19 show that the MR classifier appears to perform adequately well for the Far CP simulated data in the SNR 3:1 and 1:1 cases. The performance on SNR 0.3:1 data was not satisfactory and the next section takes a closer look at the noise sensitivity for both Hebbian and K-SVD dictionaries.

4.8 Noise sensitivity of learning methods

The dictionary size is again fixed for both methods at K=256 (i.e., undercomplete dictionaries) and performance on Far CP test data sets with the three target signal to noise ratios (3:1, 1:1, and 0.3:1) is evaluated. Using SNR 3:1 training data, 10 K-SVD and 10 Hebbian ON/OFF dictionary pairs are learned and then tested against unseen SNR 3:1, 1:1, and 0.3:1 test data. The accuracy achieved is summarized in Figure 4.20. Likewise, Figure 4.21 summarizes the accuracy obtained using dictionaries learned from Far CP SNR 1:1 training data. As expected, performance degrades in both cases as the signal to noise ratio of the test data decreases (top, middle, and bottom rows of Figures 4.20-4.21). Figures 4.20-4.21 also show that dictionaries learned with the Hebbian method consistently outperform their K-SVD counterparts for both types of training data and for all three test data sets, both in terms of median accuracy, as well as learning convergence (i.e., variance across the 10 pair set).

Dictionaries learned from Far CP SNR 3:1 training data

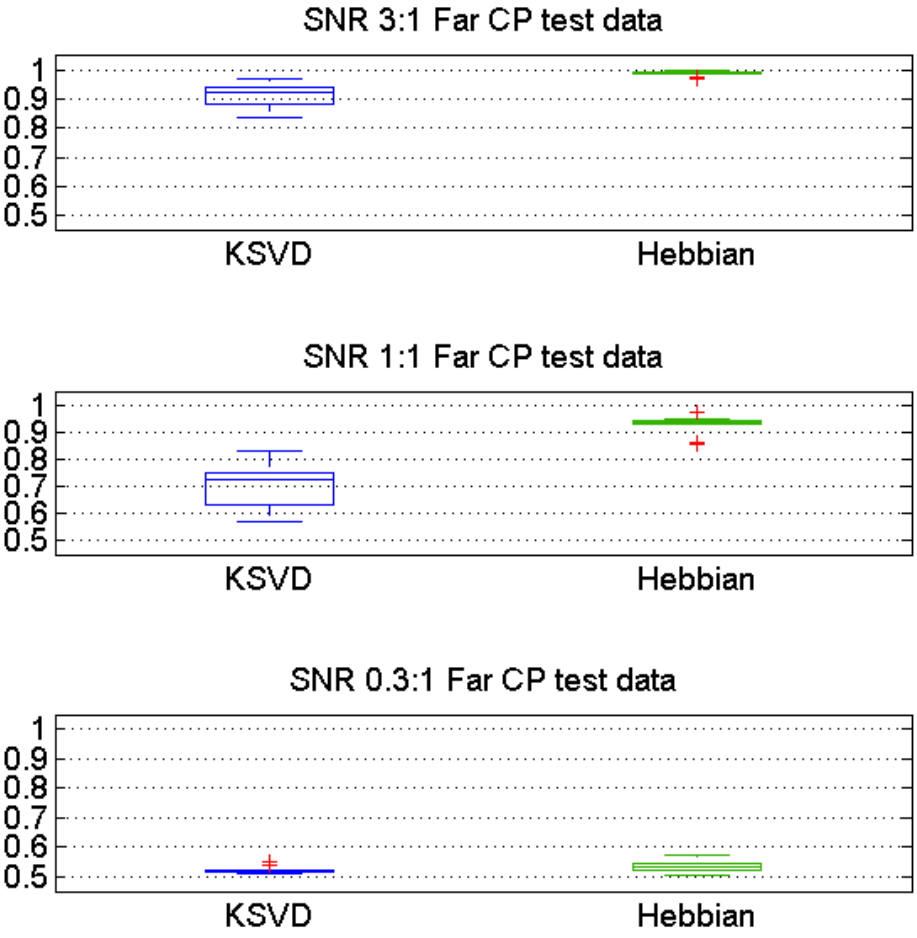


Figure 4.20: Classification accuracy rate (*y*-axis) for K-SVD and Hebbian dictionaries with K=256 elements, learned with C=25 and C=250, respectively. Dictionaries were learned on Far CP SNR 3:1 training data with sparsity factor L=45 and tested on SNR 3:1, 1:1, and 0.3:1 test data (top, middle, and bottom panels, respectively).

Dictionaries learned from Far CP SNR 1:1 training data

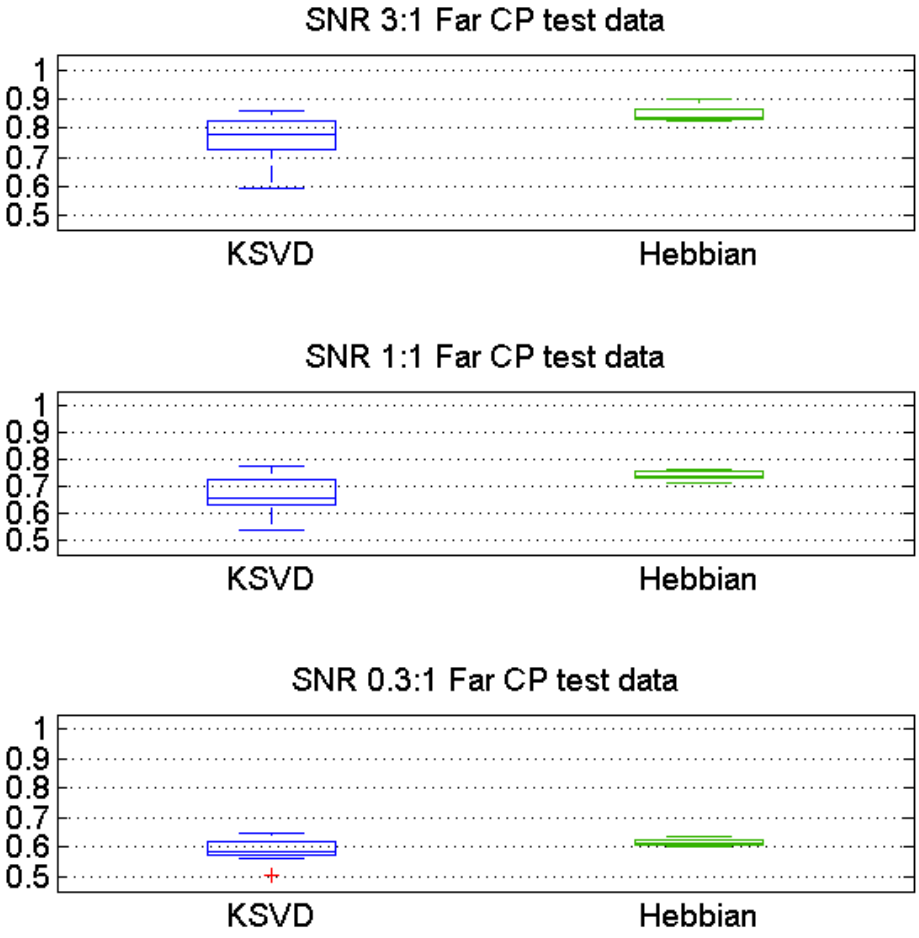


Figure 4.21: Classification accuracy rate (*y*-axis) for K-SVD and Hebbian dictionaries with K=256 elements, learned with C=25 and C=250, respectively. Dictionaries were learned on Far CP SNR 1:1 training data with sparsity factor L=45 and tested on SNR 3:1, 1:1, and 0.3:1 test data (top, middle, and bottom panels, respectively).

The top panel of Figure 4.20 shows that both dictionaries learned from SNR 3:1 training data perform well when tested on high amplitude test data. The Hebbian dictionary also performed well when tested on SNR 1:1 test data. Both approaches suffer however when applied to SNR 0.3:1 test data (bottom panel of Figure 4.20). In contrast, the panels of Figure 4.21 show that dictionaries learned from SNR 1:1 training data perform less well on high and mid amplitude test data, but they do a better job with SNR 0.3:1 amplitude test data (compare bottom panels of Figures 4.20-4.21). The sub-optimal performance of dictionaries learned from SNR 1:1 data (when applied to test data with mid and high target amplitude) may be a function of the sparsity factor or the amount of training data used in learning.

What is noteworthy in Figures 4.20-4.21 is that the classification performance is directly dependent on how "loud" the target signal is, regardless of the learning method. In other words, even when the dictionary is learned from SNR 1:1 training data, its performance improves on test data that has SNR 3:1. This is a potentially useful property of Hebbian learned dictionaries, and will be revisited in Chapter 5.

4.9 Software implementation and algorithm complexity

Recall that practical challenges for RF signal processing are the length of the time records and the short real-time processing constraints. For learned dictionary applications, the two separate algorithm components that need to be optimized are the *learning component* and the *classification component*.

Regarding the first component, learning a dictionary of size *K* can be computationally very expensive, to the point of impracticality. Since this project uses supervised learning and dictionaries are not updated with every new test set, this becomes upfront overhead and can be reduced by use of parallel computing hardware. Results show that undercomplete Hebbian dictionaries consistently outperform K-SVD dictionaries with the same size *K* for training data chosen as specified; however, this improvement in performance comes at an increase in computational cost. For Hebbian learning, the minimum number of learning iterations, *C*, for the amount of training data allotted here, needs to be roughly the same as the number of dictionary elements in order to learn a highly discriminant Hebbian dictionary. This large number of required learning iterations carries a computational burden that can be much higher than for the K-SVD method.

At a particular sequential update iteration, the *K* inner products between data and dictionary elements can be scattered across multiple cores, resulting in a computational complexity of order *O(LNP)*, where *L* is the sparsity factor, *N* is the length of a dictionary element, and *P* is the number of training data windows. In addition, parallel processing is used to learn the set of 10 pairs at the same time across different clusters. The SVD decomposition in the dictionary update step is the computational bottleneck of the K-SVD method. For example, using a parallel implementation on a Windows workstation with 8 Intel Xeon X5550 2.67GHz quadcore processors (32 cores total), it can take up to 6s per dictionary element update at every learning iteration, depending on how many training vectors are used

in the particular SVD update. Furthermore, the computational time for learning a set of 10 ON/OFF K-SVD dictionary pairs with K=256 elements and C=25 learning iterations is on average 49 minutes on the same architecture. To achieve similar accuracy using Hebbian learning on the same training data for the same dictionary size, approximately C=250 iterations are required, which takes 161 minutes per 10-pair set on the same parallel architecture.

The second algorithm component, that is, the classification of test data, can be done in almost real time using a parallel, vectorized implementation. Unlike overcomplete dictionaries, whose number of elements can be larger by an order of magnitude compared to the length of the data, N, the learned dictionaries introduced in this chapter can be *undercomplete*, i.e., $K << N$, which leads to an increase in classification speed. At each matching pursuit iteration the calculation of the K inner products between the data windows of size N and the dictionary elements can be scattered across multiple cores, reducing the complexity to $O(LT/N)$, where T is the sample length of a timeseries. If the test data is buffered (i.e., data is passed to the classifier in vectorized format of M windows on length N samples), the communication time is lowered, and further improvement is achieved from the algorithmic parallelism. Additional optimization can be achieved by distributing the calculation across the cores of a graphical processor unit (GPU), and further acceleration is possible on a GPU-accelerated cluster.

While such methodologies are not the focal point of this thesis, it is worth noting that identifying and leveraging the advantages of parallel processing is both a

novel and necessary step toward practical utilization of the classification schemes introduced over the course of this project.

4.10 Conclusion on learning dictionaries for RF classification

With learned dictionaries, the choice of the number of elements to represent a data set is still important, but the performance degrades more gradually away from the optimal number of elements. This means the learned dictionary approach is more robust to poor choices of the number of elements, and can therefore make fewer *a priori* assumptions about the data characteristics.

Two dictionary learning methods are compared in Chapter 4 for a fixed, small-size training data set. The results show that optimally-learned Hebbian dictionaries can have higher discriminative power than K-SVD dictionaries, and be more robust to small changes in SNR. For the K-SVD learning method, *undercomplete dictionaries performed as well as complete or overcomplete dictionaries in classification*. Also, it was shown that *Hebbian undercomplete dictionaries outperform K-SVD learned dictionaries of any level of completeness*. Furthermore, a hybrid learned dictionary was introduced, with faster learning convergence and more asymptotically stable behavior than a Hebbian dictionary of same size. Chapter 5 will demonstrate that a hybrid dictionary has better noise robustness as well. The question now is to address how undercomplete Hebbian dictionaries, both simple (i.e., strictly hebbian), and hybrid, will perform on the two more complex datasets, Flat CP and Chirped CP.

5. RF Classification with Undercomplete Learned Dictionaries

In Chapter 4 the focus was on the learning algorithms and their parameters, which were explored in terms of classification performance on the Far CP dataset, containing a less complex target background (i.e., a non-overlapping chirping clutter pulse). The findings of Chapter 4 are now used in Chapter 5 to optimally learn dictionaries for the more difficult cases of target detection in Flat CP and Chirped CP data, which contain competing chirped clutter within the spectral band of the target (the reader is referred to Section 1.1 for a review of these datasets and their composition). The focus of this chapter is exploring discrimination between ON and OFF target windows for these two cases using the MR classifier, under changing noise conditions. This simple classifier, introduced in Chapter 4, is selected so that the effect of the sparse representations and the learned dictionaries on classification accuracy can be readily observed; a more complicated classifier could likely improve accuracy, but it would introduce additional variables that are beyond the scope of this thesis. The flow of this chapter is as follows: a number of carefully chosen tests are performed and graphically examined to reveal characteristics (i.e., trends in performance as a function of one or more parameters) of the chosen classification scheme. These comparisons are necessarily tedious and will be summarized concisely at the end of the chapter.

A conclusion from Chapter 4 is that classification accuracy depends strongly on the number of learning iterations, *C* (i.e., amount of training). Consequently, the training set size is increased 10-fold in this chapter to 17000 ON and 17000 OFF data windows, while the test set is kept the same size. Also, *only undercomplete Hebbian dictionaries are considered in this chapter, both simple and hybrid*, given their superior performance over K-SVD in the previous section. Learning in both Flat CP and Chirped CP cases is done from SNR 3:1 respective training data, and the same MR classifier is used to discriminate between unseen test windows. As before, classification accuracy is the chosen performance metric for the work in this chapter. The behavior of classification performance with respect to learning parameters is found to be similar to the Far CP dataset, i.e., some generalization can be made. The impact of noise on classification accuracy is explored in greater detail in this chapter, and is compared to noise sensitivity of a different classifier using STFT-features.

5.1 Study of method parameters

Given that the training set has increased in size by an order of magnitude, classification accuracy is again evaluated as a function of parameters *K*, *C*, L_{train}, and L_{class} over a wide range of values. In this sensitivity analysis, sets of 10 simple Hebbian dictionary pairs for every combination of (*K*, *C*, L_{train}) are learned from SNR 3:1 training data using different random dictionary seeds, for both Flat CP and Chirped CP data (hybrid dictionaries are not yet considered). All sets of 10 ON/OFF dictionaries are used to classify previously unseen test data with varying L_{class}. A total

of four undercomplete dictionary sizes are considered, K={256, 128, 64, 32}. That is, the dictionaries are undercomplete by a factor of {0.5, 0.25, 0.125, and 0.0625}.

5.1.1 Learning iterations, *C*

First, a range of values for C is considered to explore dictionary properties as a function of learning iterations using 10 sets of Hebbian ON/OFF dictionary pairs learned from SNR 3:1 training data. Based on the results in Chapter 4 and given the increased amount of training data, the expectation is that asymptotic accuracy performance as a function of learning iterations would be reached sooner than the C values observed in Chapter 4 (e.g., ~250 iterations for K=256).

The error-bar plot in Figure 5.1 shows resulting classification accuracy for Flat CP dictionaries of the four specified sizes for SNR 3:1 test data (top plot) and SNR 1:1 test data (lower plot). The other parameters are kept equal and constant for each K, specifically L_{train}=L_{class}={45, 36, 15, 8} for K={256, 128, 64, 32}. Similarly, Figure 5.2 shows the equivalent SNR 3:1 and SNR 1:1 cases for the Chirped CP test data case. The error-bar plots show median classification accuracy (solid traces) obtained over the respective set of 10 pairs, and its standard deviation (vertical bars). Good accuracy performance would be indicated not only by high median accuracy, but also by very small respective standard deviation (i.e., similar data space representation fidelity across the 10-pair set).

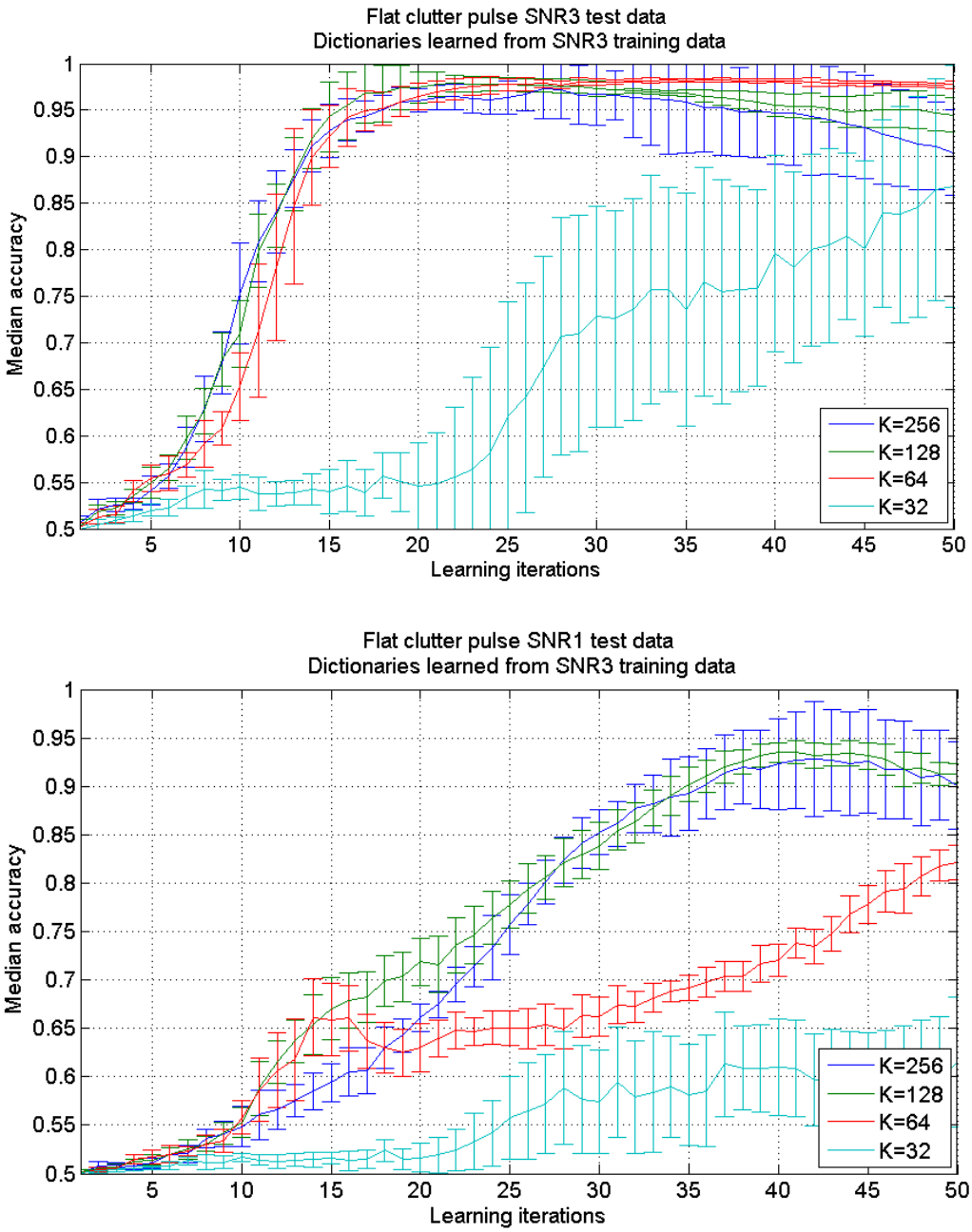


Figure 5.1: Error-bar plots of classification accuracies over the 10 pair dictionary set for a range of learning iterations, C, in the Flat CP SNR 3:1 test data case (top) and SNR 1:1 case (bottom). Dictionaries are learned from SNR 3:1 training data. Here L_{train}=L_{class}={45, 36, 15, 8} for K={256, 128, 64, 32}.

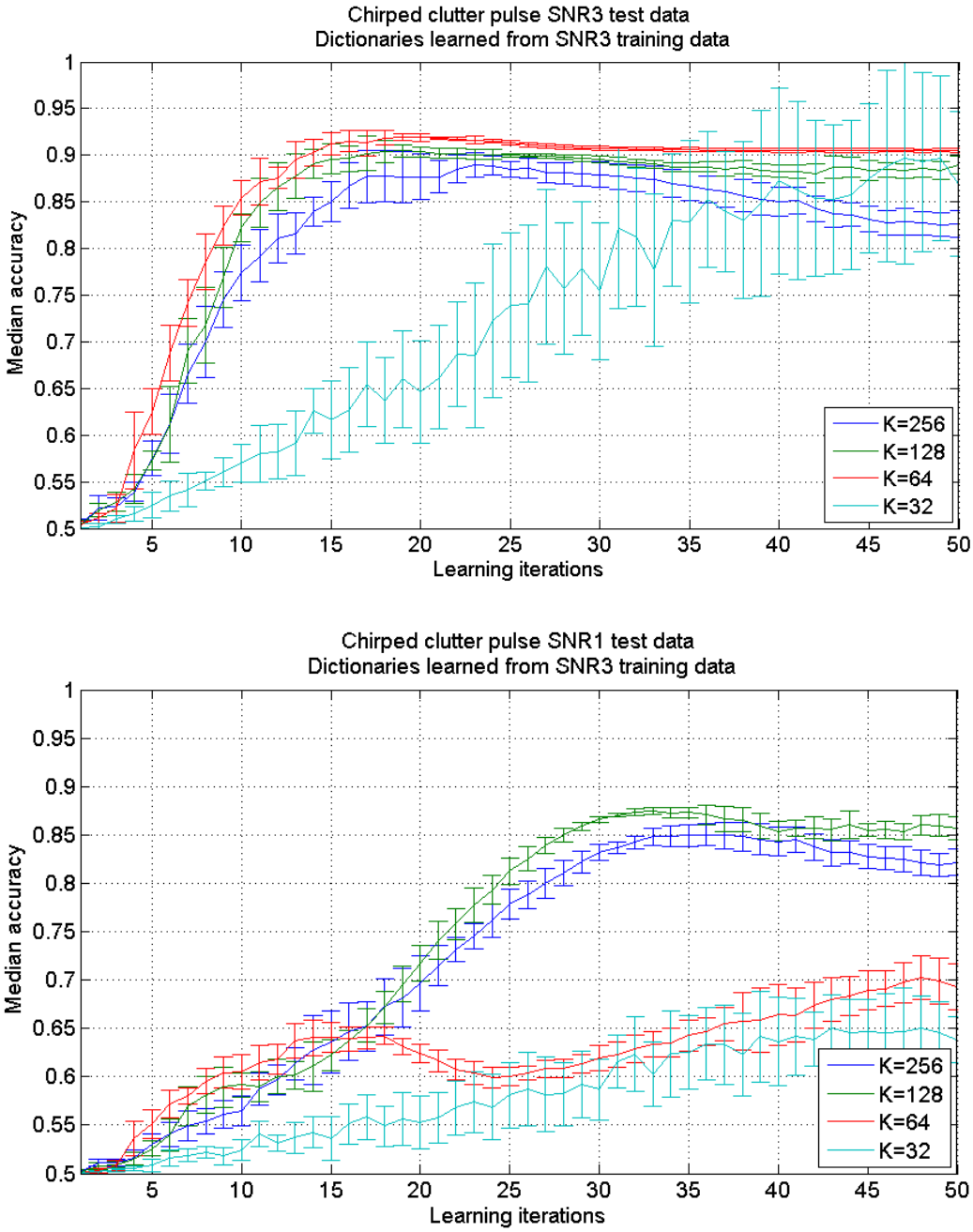


Figure 5.2: Error-bar plots of classification accuracies over the 10 pair dictionary set for a range of learning iterations, C, in the Chirped CP SNR 3:1 test data case (top) and SNR 1:1 case (bottom). Dictionaries are learned from SNR 3:1 training data. Here L_{train}=L_{class}={45, 36, 15, 8} for K={256, 128, 64, 32}.

In Figure 5.1, for Flat CP SNR 3:1 test data (top panel), very good classification (median accuracy ~0.97, peak accuracy ~0.995) can be obtained with undercomplete dictionaries of size $K \geq 64$ with sufficient learning iterations, i.e., sufficient training. All $K \geq 64$ dictionaries reach their peak median accuracy around C~20-25 learning iterations. If the dictionary size is too small (e.g., $K=32$), the classification accuracy is poor, exhibits high variation, and does not reach convergence. For $K=64$ and 128, the median accuracy first exhibits a sharp increase, followed by a small region of peak performance, and then by a slight decay down to a relative asymptotic performance. For $K=256$, after the peak performance region, median accuracy continues to decay and does not appear to stabilize for the number of learning iterations explored. This decay in performance for large number of learning iterations could be due to overfitting the training data in the learning stage; this hypothesis is supported by the increasingly larger variance across the 10 pair set.

In the Flat CP SNR 1:1 test data case (bottom panel of Figure 5.1), it would appear that the $K=64$ dictionary is now also "too small" and behaves similarly to the $K=32$ case. The best median performance of ~0.935 is reached by $K=128$ dictionaries, which also have smaller variance across the 10 pair set. The best individual peak performance of ~0.985 accuracy is reached by a $K=256$ dictionary with $C=42$ iterations.

For Chirped CP data (Figure 5.2), the overall classification performance is slightly worse, as expected for this case. The behavior of dictionaries with size $K \geq 64$ is very similar in the SNR 3:1 test data case for both Flat CP and Chirped CP data.

Figure 5.2 shows again asymptotic classification behavior for K=64, i.e., after ~25 learning iterations the median performance flattens out and there is very little variance across the 10 pair set, while for K=256 the same pattern of median performance worsening is observed. Similar to the top panel of Figure 5.1, for SNR 3:1 Chirped CP data, the $K \geq 64$ dictionaries reach their peak median accuracy (~0.92) between C~20-25 learning iterations.

For SNR 1:1 Chirped CP test data (Figure 5.2, lower panel), median accuracy patterns resemble those observed in the SNR 1:1 Flat CP case (Figure 5.1, lower panel). The K={128, 256} dictionaries show more constant and smaller variability as C increases, and reach their peak performance sooner (32-35 iterations) than dictionaries with K={32, 64} (40-45 iterations).

In the top panels of Figures 5.1 and 5.2, peak performance is reached with a lower number of learning iterations for SNR 3:1 test data (C=15-20), but that amount of learning is nowhere near sufficient for SNR 1:1 test data (lower panels of Figures 5.1 and 5.2). The amount of learning necessary for peak performance with a particular dictionary size for SNR 1:1 test data (e.g., C~42 for Flat CP data and K=256) is not necessarily in the optimal learning region for SNR 3:1 test data with the same dictionary size. This presents a problem, as the goal is to learn dictionaries that are robust to changes in noise conditions. In Section 5.2, a solution will be given using the concept of novel hybrid dictionaries introduced in Chapter 4.

5.1.2 Learning sparsity factor, L_{train}

In Chapter 4, the effects of the sparsity factor in learning (see Section 4.5.3), L_{train}, and in classification, L_{class}, were separately explored for Flat CP data. It was shown that L_{class} could have smaller values than L_{train} without loss of classification performance compared to L_{class}=L_{train}, for high enough L_{train} values (Figure 4.17).

A central hypothesis in this dissertation is that perfect reconstruction is not necessary for classification with learned dictionaries, i.e., for learning good discriminative features. This results in many advantages, as discussed in this chapter and those that follow. The accuracy performance as a function of L_{class} and L_{train} observed in Chapter 4 was a significant step towards supporting that hypothesis, as these sparsity factors directly impact the reconstruction error, i.e., they control how well the data is approximated by the sparse combination of dictionary elements. Also, in Figure 4.12, the classification performance became relatively stable as early as L_{train}>8 (with L_{class}=L_{train}), which was considerably lower than the relatively high value of L_{train}=45 that was predominantly used in Chapter 4. In other words, the classification performance obtained with coarser approximations (i.e., smaller L_{train}), was the same as that obtained with finer approximations, but without the added computational cost.

The focus of this section is to explore in greater detail these effects for undercomplete dictionaries, in particular how coarse the approximations can be during the learning process for the dictionary to still learn good discriminative features. A range of learning sparsity factors, L_{train}, is considered for a fixed number

of learning iterations, C=33, and two fixed classification sparsity factors L_{class}. Dictionaries learned with C=33 are selected based on Figures 5.1 and 5.2 for being out of the transitional region (i.e., for small C) of accuracy performance, leading to relatively high accuracy for all four dictionary sizes, and, more importantly, for exhibiting consistent standard deviation in their neighborhood of learning iterations. Sets of 10 simple Hebbian ON/OFF dictionary pairs are learned with L_{train} ={4:4:32} from SNR 3:1 training data. Using the MR classifier, unseen SNR 3:1 test data is classified using first L_{class}=32, and secondly L_{class}=8. For this discussion, the focus is on Chirped CP data, which is the more challenging case to classify as the target and clutter pulses are more similar. The author notes that the classification accuracy behavior was extremely similar in the Flat CP case, and higher peak accuracy was observed, as expected. Figure 5.3 shows error bar plots of median accuracy as a function of L_{train} for fixed L_{class}=32 for all four dictionary sizes, and Figure 5.4 similarly shows accuracy for fixed L_{class}=8.

In Figure 5.3, for the values considered, dictionaries with $K\geq$64 appear insensitive to the learning sparsity factor once a minimum L_{train}=8 is reached. For K=32, accuracy peaks in the L_{train}=(8:12) region, followed by slow degradation. The highest median (~0.92) and individual peak (~0.96) accuracy is in this case reached for K=32 dictionaries around L_{train}~8. Since classification is made with L_{class}=32, that is, $L_{class}>L_{train}$, these results represent classification using a finer approximation than the approximation used in training. It can be seen from Figure 5.3 that by using a finer approximation in classification (larger L_{class}), the degree of coarseness in

learning (smaller L_{train}) has a negligible impact on final accuracy for $K \geq 64$ and $L_{train} \geq 8$.

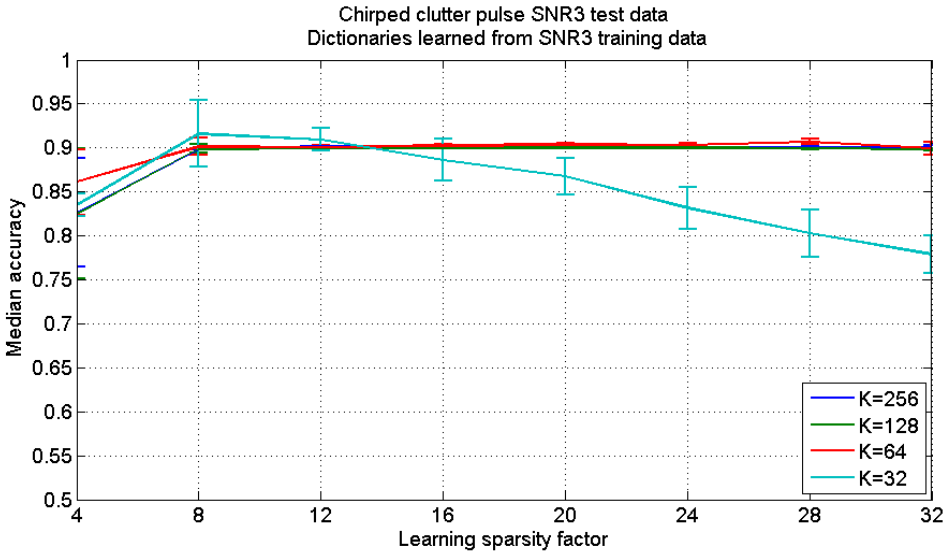


Figure 5.3: Error-bar plots of median accuracies over the 10 pair dictionary set for various learning sparsity factors, L_{train}, for the Chirped CP SNR 3:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{class}=32.

An interesting question is raised by the performance at L_{train}=8 in Figure 5.3 for all four dictionary sizes. At such learning sparsity factor, the approximation is quite coarse and the residual decay for this dataset is generally in its roll-off region (similar to Figure 4.11). To explore this further, the classification sparsity factor is selected to be L_{class}=8 and L_{train} is again varied in the {4, 32} interval. Figure 5.4 shows the accuracy plots corresponding to MR classification made with a coarser approximation (i.e., fewer dictionary elements in the sparse representation during classification). Here the trends are similar to Figure 5.3, but the classification exhibits more variability (i.e., larger deviations).

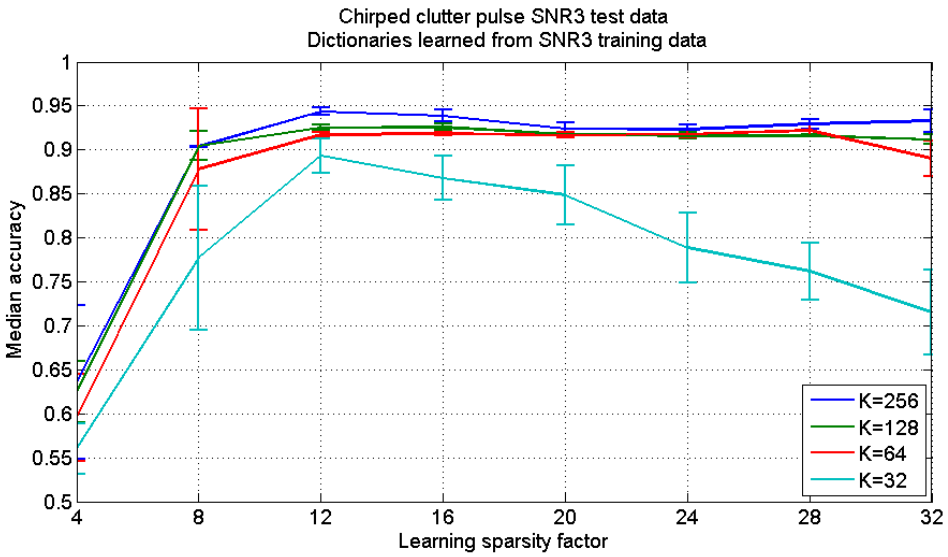


Figure 5.4: Error-bar plots of median accuracies over the 10 pair dictionary set for various learning sparsity factors, L_{train}, for the Chirped CP SNR 3:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{class}=*8*.

Higher median peak performances are reached by all dictionaries with $K \geq 64$ using coarser approximations in classification (Figure 5.4), but that higher performance does not appear quite as stable (i.e., as flat) as it did in the L_{class}=32 case (Figure 5.3), except for K=64 in the L_{train}={12:28} region. However, even with the slight variations, the median performance for dictionaries with $K \geq 64$ in the L_{train}={12:28} is consistently higher than the median ~0.9 performance observed in Figure 5.3. The improvement in accuracy performance from L_{train}=4 to L_{train}=8 is much more abrupt, and in the K=32 case the performance decay after the peak has a much steeper rate. Similarly, the minute performance decay for K=64 in Figure 5.3 after L_{train}=28 becomes more pronounced in Figure 5.4.

Recall that the goal of this section is to evaluate classification performance using coarse approximations. Figure 5.4 indicates that better classification

performance can be obtained by selecting a coarse approximation in the learning stage (e.g., L_{train}=12) and using a similarly coarse or coarser approximation in classification (e.g., L_{class}=8). This result is consistent with that of Figure 4.12, which showed that the accuracy dependence of L_{train} for the simpler Far CP data enters an asymptotic region for L_{train}>12 (and in that case L_{class}= L_{train} =12). Using coarser approximations has effectively improved median peak accuracy by 0.05 (compare Figures 5.3 and 5.4). It is also useful to reduce computational overhead in the learning and classification stage, since it implies computing a coarser approximation (i.e., smaller number of dictionary inner products and matching pursuit searches).

5.1.3 Classification sparsity factor, L_{class}

Lastly, the changes in accuracy introduced by varying the classification sparsity factor, L_{class}, are explored for Chirped CP dictionaries learned from SNR 3:1 data. The number of learning iterations is again kept constant at C=33, and two learning sparsity factors are considered: first L_{train}=32, and secondly L_{train}=12. Performance is evaluated both for SNR 3:1 test data, as well SNR 1:1 test data. Recall that the previous section showed learning the dictionary from coarse approximations is actually beneficial in classification, but only hinted that a coarse approximation in classification might also be useful. The goal of this section is to now fully explore whether using coarse approximations in the classification stage is a viable option for good classification. Figure 5.5 shows error bar plots of median accuracy on SNR 3:1

test data, as a function of L_{class}, where all four dictionaries were learned with L_{train}=32.

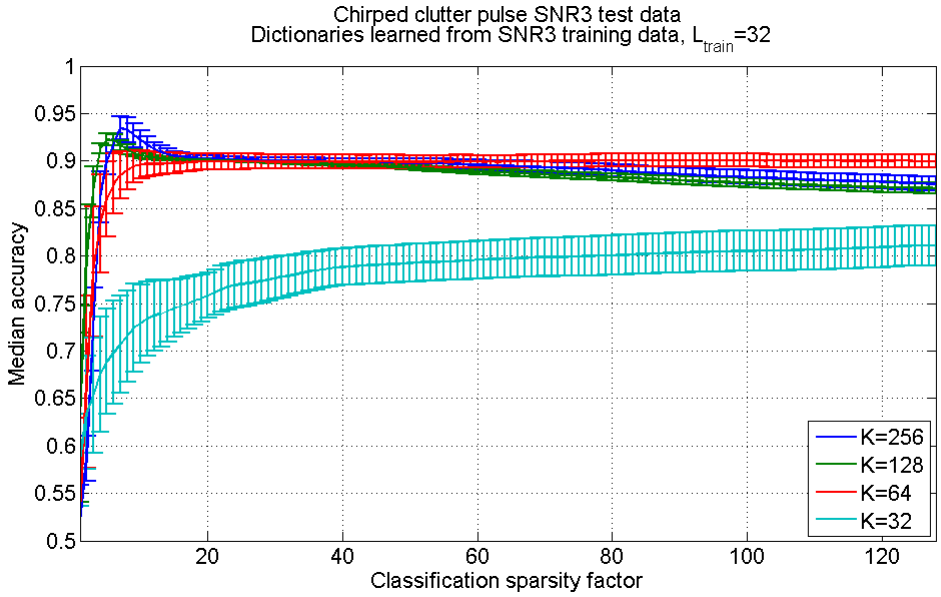


Figure 5.5: Error-bar plots of median accuracies over the 10 pair dictionary set for various classification sparsity factors, L_{class}, for the Chirped CP SNR 3:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{train}=32.

For dictionaries with $K \geq 64$ accuracy peaks by L_{class}=10. For values of L_{class} > 20, accuracy stays relatively constant for K=64 and slowly degrades for K={256, 128} as L_{class} increases up to 128. For values of L_{class}<128, dictionaries with K=32 continually improve in accuracy as L_{class} grows, but their median accuracy remains lower than the accuracy obtained with larger dictionaries. Using an $L_{class}>K$ value is made possible by the matching pursuit approach employed, which searches the dictionary with replacement, i.e., the same dictionary element can be selected as the best match multiple times. Comparing Figure 5.5 with Figure 5.3, the median accuracies with L_{train}=32 of ~0.9 can be again recognized here at L_{class}=32, and

similarly for Figure 5.4 and L_{class}=8, median accuracies with L_{train}=32 are ~0.93. The best performance is again obtained by dictionaries with K≥64, and, consistent with the conclusion of the previous section, for coarser approximations in classification (L_{class}<10).

Figure 5.6 similarly shows error bar plots of median accuracy obtained when classifying SNR 1:1 Chirped CP data, as a function of L_{class}, for all four dictionary sizes learned with L_{train}=32.

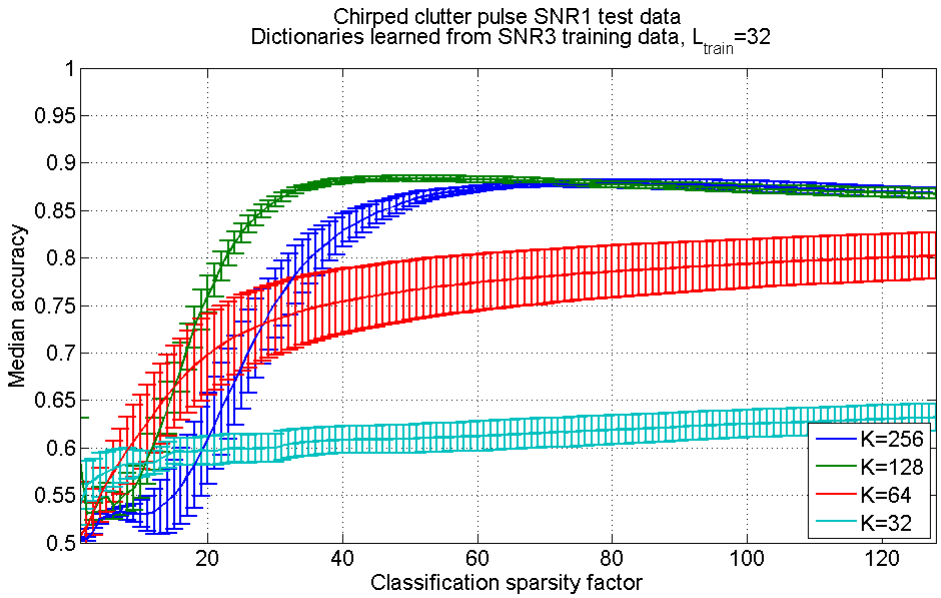


Figure 5.6: Error-bar plots of median accuracies over the 10 pair dictionary set for various classification sparsity factors, L_{class}, for the Chirped CP SNR 1:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{train}=32.

Consistent with the classification performance pattern in Figure 5.2 (bottom plot), in the SNR 1:1 test data case dictionaries with K=64 drop in performance and resemble more K=32 dictionaries in behavior. That is, for values of L_{class}<128, dictionaries with K={32, 64} continually improve in accuracy, but their median

accuracy remains lower than the accuracy obtained with K={128, 256} dictionaries. For these latter larger dictionaries, in this SNR test regime, L_{class} needs to be higher than in the SNR 3:1 case, and is in fact higher than L_{train}=32. Peak median accuracy of ~0.88 is obtained with K=128 around L_{class}~47. In other words, noisier data needs finer approximations for asymptotic classification performance.

When reducing the learning sparsity factor to L_{train}=12 (i.e., coarser approximations are used in learning), significant performance improvements are noticeable, in particular for the very undercomplete dictionaries. Figure 5.7 shows error bar plots of median accuracy as a function of L_{class} for all four dictionaries learned with L_{train}=12. The most noticeable difference is observed for the K=32 case, which is remarkable given its degree of undercompleteness. For the values of L_{class} tested, dictionaries with K=32 still show gradual improvement in accuracy as L_{class} grows, but their median accuracy is now higher than the accuracy obtained with larger dictionaries, different from the results in Figure 5.5. They also exhibit the same sharp rise in performance for very small values of L_{class}. Compared to Figure 5.5, accuracy still peaks by L_{class}=10 for dictionaries with $K\geq$64 in Figure 5.7. For values of L_{class} > 20, accuracy stays relatively constant for K=64 and gradually degrades for K={256, 128} as L_{class} increases up to 128, at a rate similar to that of Figure 5.5**.** The best performance for dictionaries with $K\geq$64 is again obtained using coarser approximations in classification, i.e., L_{class}<10. Peak median accuracy of ~0.945 is obtained with $K\geq$256 dictionaries at L_{class}~8, similar to the results in Figure 5.4.

For all values of K tested, using a coarse learning sparsity factor leads to smaller variance across the 10 pair dictionary set in the SNR 3:1 test data case, i.e., better learning convergence.

Figure 5.8 shows error bar plots of median accuracy obtained when classifying SNR 1:1 data, as a function of L_{class}, where the four dictionaries are now learned with L_{train}=12. Unlike Figure 5.6, dictionaries with $K \geq 64$ now exhibit similar behavior to K={128, 256} dictionaries, and reach similar accuracy levels (~0.86 median accuracy). Dictionaries with K=32 continually improve in accuracy for L_{class}<128, but their median accuracy remains lower than the accuracy obtained with K={64, 128, 256} dictionaries. It is notable that with the coarser learning approximations (Figure 5.8), K=32 dictionaries improve up to median accuracies of 0.7, whereas in the finer learning approximation case (Figure 5.6), the best median accuracy was 0.63 for K=32. Peak median accuracy of ~0.875 is obtained with K=128 around L_{class}=60. Similar to Figure 5.6, for the larger dictionaries in the SNR 1:1 test regime, L_{class} again needs to be higher than for the SNR 3:1 case (e.g., 40-60), or in other words, noisier data requires finer approximations in classification. As previously noted in Figure 5.7, coarser learning approximations lead to smaller variances across the 10 pair dictionary set, which is important from a standpoint of learning convergence, or algorithm stability.

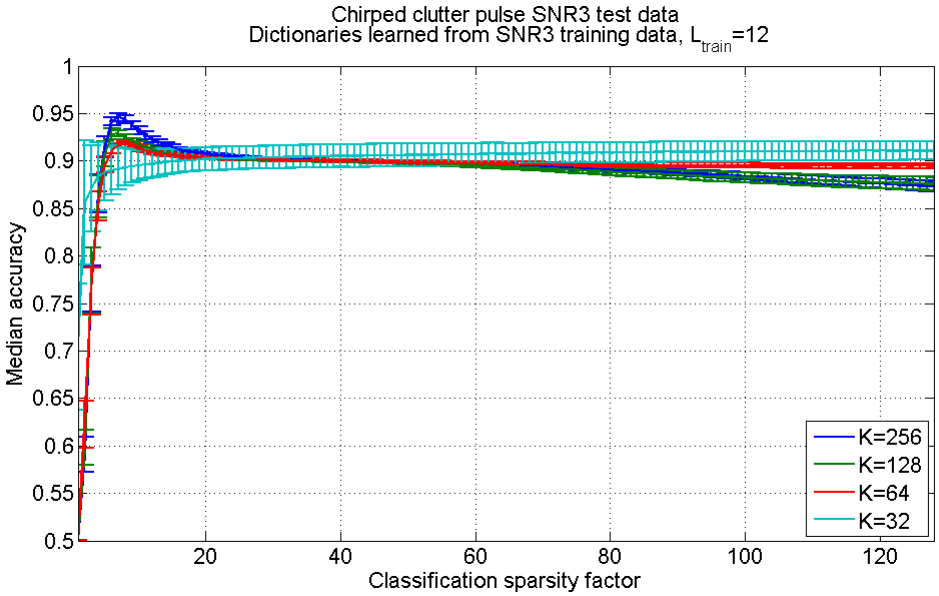


Figure 5.7: Error-bar plots of median accuracies over the 10 pair dictionary set for various classification sparsity factors, L_{class}, for the Chirped CP SNR 3:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{train}=12.

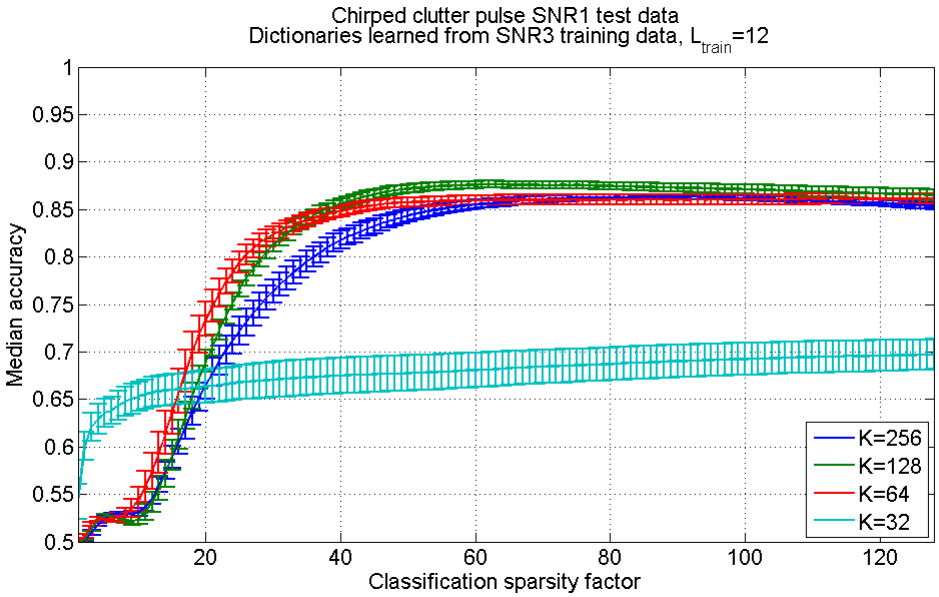


Figure 5.8: Error-bar plots of median accuracies over the 10 pair dictionary set for various classification sparsity factors, L_{class}, for the Chirped CP SNR 1:1 test data case. Dictionaries are learned from SNR 3:1 training data. Here C=33, and L_{train}=12.

Figures 5.1-5.8 show that lower bounds exist for some parameter values in order to reach good classification accuracy for the RF simulated data. These lower bounds can be generalized and lead to methodologies for refining a specific classification scheme. If the dictionary size K is too undercomplete (i.e., see cyan curves for $K = 32$), classification accuracy suffers greatly. Similarly, all the parameters show some minimum level below which classification accuracy is unacceptable. For example, $L_{train} < 8$, $L_{class} < 12$, and $C < 15$ are regions that indicate minimum values for reasonable performance on the specified datasets. Furthermore, once that minimum parameter setting is reached, the panels demonstrate that performance remains approximately constant (or shows slow degradation) as the parameter settings increase for some optimal interval. Contrary to a reconstruction-driven application, in this classification scenario lower value parameters in the optimal interval are preferable, i.e., "just enough" learning iterations, and "just enough" approximation coarseness. As for dictionary size, the detailed study of Section 5.1 showed that, for some given training set, a wide range of undercomplete dictionaries (e.g., with as low as 0.125 degree of undercompleteness) can be optimally trained to yield high classification performance over a relatively broad interval of algorithm parameters .

Since learned dictionaries require many *a priori* decisions regarding parameter settings, it is encouraging to know that, even with the number of training examples used in the current sensitivity analysis, satisfactory results were achieved across a range of parameter settings. This suggests that a practitioner applying a

learned dictionary approach could identify the minimum parameter settings that optimize performance on the training data (using an approach similar to the one followed above) and achieve reasonably robust results on unseen test data. The importance of this conclusion should be emphasized, as it confirms immediate feasibility in realistic applications (discussed in the next chapter).

5.2 Hybrid dictionaries performance

Hybrid dictionaries were introduced in Chapter 4 as a novel solution to a.) speeding up convergence for Hebbian dictionaries in terms of required number of learning iterations for a fixed size training set, and b.) providing a more stable classification performance (i.e., relatively constant accuracy performance over a wider range of parameters). Hybrid dictionaries are now learned with just 1 K-SVD seed in sets of 10 pairs from Flat CP and Chirped CP SNR 3:1 training data, using the larger training set size of 17000 ON and 17000 OFF windows. Only three dictionary sizes are considered here, K={256, 128, 64}, based on classification performance observed in Section 5.1, each with fixed sparsity factors of L_{train}=L_{class}={45, 36, 15}, respectively. The values for the sparsity factors were chosen to be the same as those in Figures 5.1-5.2 to allow accuracy dependence on number of learning iterations to be directly compared. The SNR 3:1 test data case is considered first for both data types. The median accuracy plots in the top panels of Figures 5.1 and 5.2 are now replicated again (without the standard deviation bars), this time to include the equivalent Hybrid performances (Figures 5.9 and 5.10).

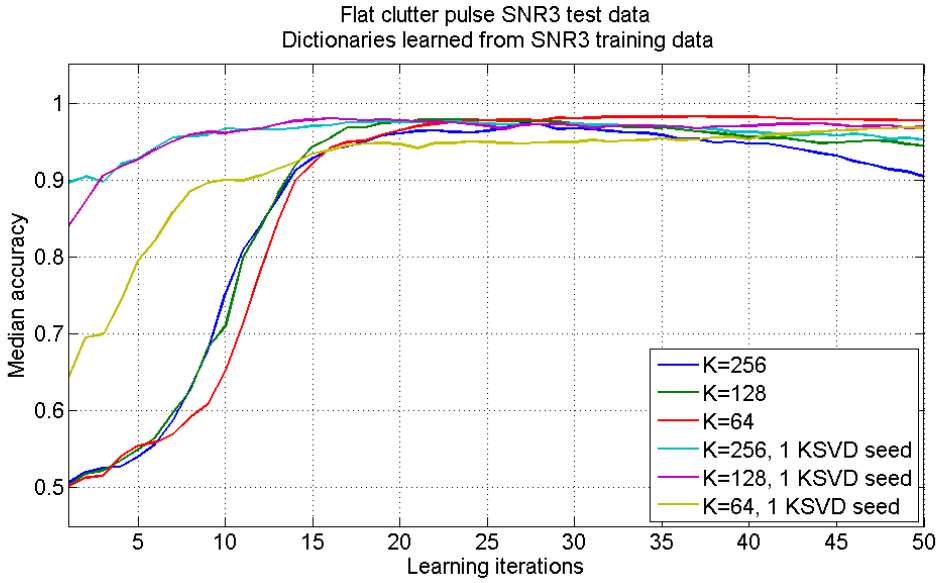


Figure 5.9: Classification accuracy using Hebbian learned dictionaries of sizes K={256, 128, 64} for Flat CP SNR 3:1 test data. A range of C=1 to C=50 learning iterations was used to learn sets of 10 ON/OFF dictionary pairs. Both Hebbian dictionaries initialized with random uniform seed, as well as Hebbian dictionaries initialized with 1 K-SVD seed were used. Median performance is shown across the set of 10 pairs as a function of the number of learning iterations.

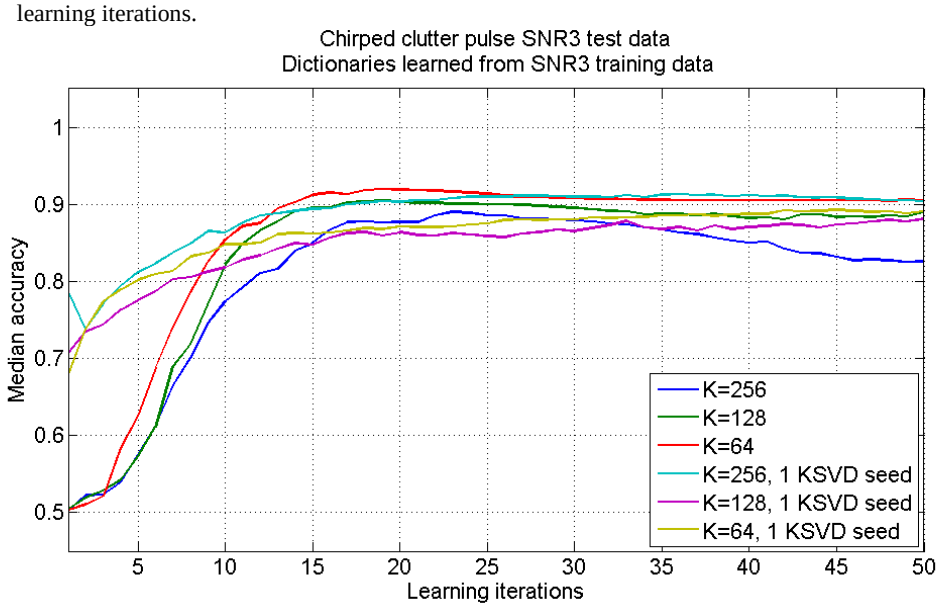


Figure 5.10: Classification accuracy using Hebbian learned dictionaries of sizes K={256, 128, 64} for Chirped CP SNR 3:1 test data. A range of C=1 to C=50 learning iterations was used to learn sets of 10 ON/OFF dictionary pairs. Both Hebbian dictionaries initialized with random uniform seed, as well as Hebbian dictionaries initialized with 1 K-SVD seed were used. Median performance is shown across the set of 10 pairs as a function of the number of learning iterations.

Accuracy reaches asymptotic behaviour by C~15 iterations for the hybrid dictionaries, and remains relatively stable until C~50 iterations, for both Flat CP and Chirped CP data. As previously seen, the accuracy reaches higher levels in the Flat CP case compared to the Chirped CP case. Comparing hybrid dictionaries with simple Hebbian dictionaries of various sizes, Figures 5.9 and 5.10 show that simple dictionaries are more prone to overfit the training data as the number of learning iterations increases, leading to decay in performance for higher values of C. For the individual dictionary sizes, the comparsion between hybrid and simple dictionaries is distinct for K={128, 256} and K=64 cases. In Figures 5.9 and 5.10, the median accuracy for simple dictionaries with K={128, 256} (green and blue traces) rises sharply until C=15 and peaks around C~20-25 iterations, followed by a gradual decay which is more pronounced in the K=256 case (blue trace). Hybrid dictionaries with the same sizes (magenta and cyan traces) exhibit a much higher accuracy from the first few iterations, and retain a more flat performance across a wider range of learning iterations (i.e., are more stable). This performance stability will prove very useful in the case of SNR 1:1 test data, as will be seen shortly. For the K=256 case, hybrid dictionaries (cyan trace) perform better than their simple counterparts (blue trace) in terms of classification accuracy, for both Flat CP and Chirped CP data. For dictionaries with K=64 elements, the improvement in accuracy for the hybrid case (yellow trace) is still faster than the simple case (red trace) for the first few learning iterations, but now the peak accuracy is no longer obtained by the hybrid dictionaries. Among the simple dictionaries, K=64 was the only case demonstrating asymptoptic

accuracy performance in both Figure 5.9 and 5.10, and this accuracy is not surpassed by its hybrid counterpart. One observation is that the accuracy for the K=64 hybrid dictionary continues to improve for higher values of C, and becomes close to the asymptotic performance of the simple K=64 dictionary case. It appears that the larger the undercomplete dictionary (e.g., K=256,128), the more prone it is to "overlearn" (overfit) training data as the number of learning iterations increases, resulting in decrease of classification performance when applied to test data. In contrast, smaller undercomplete dictionaries (e.g., K=64) show stable performance or improvement over the range of learning iterations, likely due to the fact that their smaller number of elements preclude detailed learning of all the variability present in the training data.

The hybrid dictionaries learned from SNR 3:1 data are now applied to SNR 1:1 test data for both Flat CP and Chirped CP cases. Figures 5.11 and 5.12 replicate the bottom panels of Figures 5.1 and 5.2, enhanced with the equivalent hybrid performances, and excluding standard deviation bars. As previously noted, the initial accuracy for the hybrid dictionaries is higher, and the progression to best accuracy as a function of learning iterations is more gradual compared to the simple counterparts. Among the three different dictionary sizes explored, both simple and hybrid, peak performance is obtained for the hybrid case with K=256 (cyan traces) for both Flat CP and Chirped CP cases. In the Flat CP case, the peak median accuracy of 0.946 is obtained at C=44 learning iterations. In the Chirped CP case, the peak median accuracy of 0.876 is obtained for C=49 iterations, with a close second of 0.874 obtained with K=128 (simple case – green trace) at C=33 learning iterations.

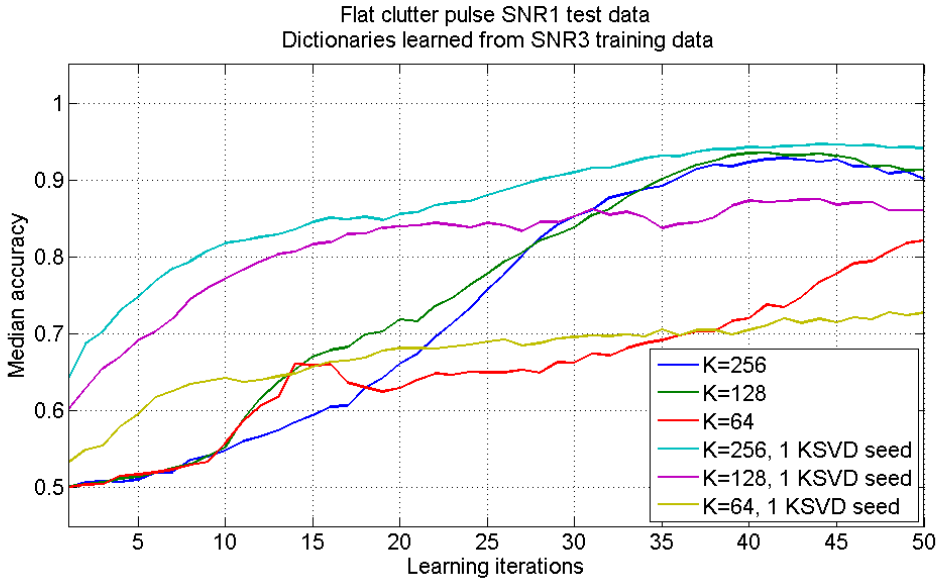


Figure 5.11: Classification accuracy using Hebbian learned dictionaries of sizes *K*={256, 128, 64} for Flat CP SNR 1:1 test data. A range of C=1 to C=50 learning iterations was used to learn sets of 10 ON/OFF dictionary pairs. Both Hebbian dictionaries initialized with random uniform seed, as well as Hebbian dictionaries initialized with one K-SVD seed were used. Median performance is shown across the set of 10 pairs as a function of the number of learning iterations.

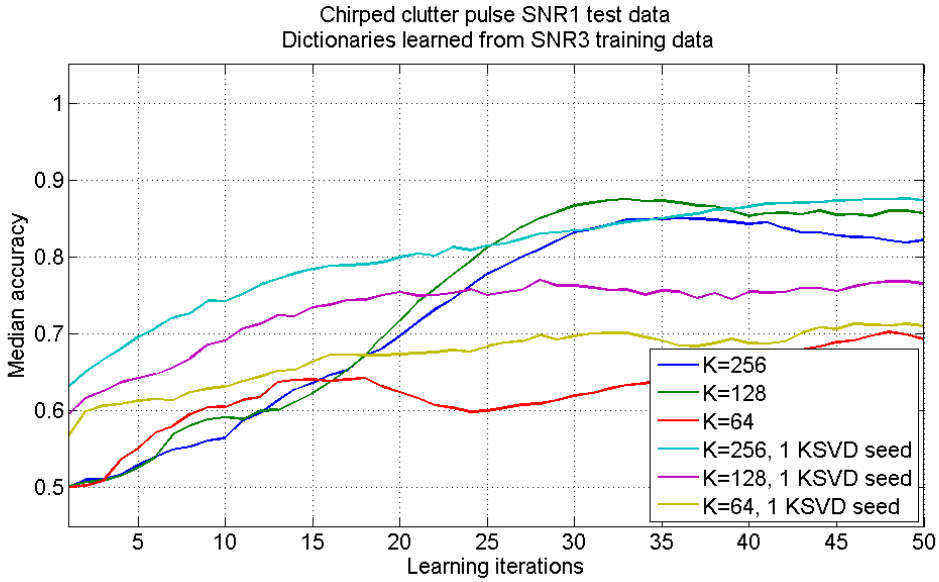


Figure 5.12: Classification accuracy using Hebbian learned dictionaries of sizes *K*={256, 128, 64} for Chirped CP SNR 1:1 test data. A range of C=1 to C=50 learning iterations was used to learn sets of 10 ON/OFF dictionary pairs. Both Hebbian dictionaries initialized with random uniform seed, as well as Hebbian dictionaries initialized with one K-SVD seed were used. Median performance is shown across the set of 10 pairs as a function of the number of learning iterations.

For K={128,64} peak median accuracy performance in the SNR 1:1 test data case is worse in the hybrid case than the simple case. For both data cases, accuracy for the respective hybrid K=256 dictionary is showing similar flat performance on SNR 1:1 test data in the range of C~40-50 learning iterations. In Figures 5.9 and 5.10, the same dictionary exhibits a relatively asymptotic, high accuracy performance, for SNR 3:1 test data in the same range of C~40-50 learning iterations (with very minor degradation in the Flat CP case). This is an encouraging result, as the performance in both noise scenarios is made optimal for the same dictionary parameters, thus addressing the concerns at the end of Section 5.1.1.

Using hybrid dictionaries can be useful in applications where an extensive sensitivity analysis on the learning parameters (similar to the one carried out at the start of this chapter) may not be possible due to practical constraints. In this thesis, the K-SVD algorithm was used to provide the seed for learning the Hebbian dictionary, but other sparsifying transforms on the data could be also employed to provide the dictionary seed. For undercomplete dictionaries, the hybrid algorithm provides faster learning convergence and more stable asymptotic behavior with respect to classification accuracy. Recall that on a different training set size, and even for a complete dictionary (i.e., K=512), the same pattern for accuracy with hybrid dictionaries was observed in Chapter 4. Given the mathematical formulation of both learning algorithms, such initial faster accuracy improvement can be expected in general of hybrid dictionaries, as it is a direct effect of initialization with a "first-glance" at the data via the K-SVD seed. After some region of relatively flat hybrid

performance, as learning iterations increase it is conceivable that the hybrid dictionary would also start overfitting the training data at some point, as is the case with learning in general.

Even though reconstruction performance is not directly assessed in this work, it is indirectly evaluated given the use of an MR classifier. However, the only conclusion on reconstruction performance that can be inferred from the classification performance is that the mean square errors over the ON and OFF dictionaries retain similar relative values; nothing can be said regarding their absolute values.

5.3 Robustness to changes in SNR

This section explores the changes in classification accuracy *when the amplitude of the target in the training set differs from the amplitude of the target in the test set.* This is an important consideration because real world applications cannot in general guarantee that these amplitudes are equal. One obvious competitor to learned dictionaries, the short-time Fourier transform (STFT), is appealing because it does not require so many *a priori* parameter settings; only the window size need be chosen ahead of time and training the classifier incurs little up-front cost. As demonstrated below, when the training and test data are drawn from the same population (i.e., have the same background characteristics), classification results based on STFT features are better or equivalent to those based on learned dictionary features.

Thus, it would seem clear that the STFT wins because it is simple and fast to implement and performs well during classification. *However, the goal of this work is to develop algorithms that perform well when applied to poorly characterized targets and backgrounds.* A key aspect is that the relative strengths of the target and background are unlikely to be known *a priori*. Therefore, the performance of the feature extraction technique must be assessed when the amplitude of the target in the test data *differs* from the amplitude of the target in the training data.

Classification performance of learned dictionaries is now compared with STFT-based classification of Flat CP and Chirped CP data in the three SNR regimes described in Section 1.1. Learned dictionaries are used with the MR classifier, and the STFT is used with a decision tree classifier from the Weka collection [53], which was selected for its superior accuracy performance on the STFT features. Decision trees are a classic method of organizing classification schemes, and offer a fast and powerful way to express structures in data. The particular algorithm used by Weka to find a decision tree is known as J48, and is a version of the earlier C4.5 algorithm developed by J. Ross Quinlan [122]. Here a decision tree is trained to map observations about an analysis window timeseries to conclusions about the window's label (ON or OFF) value. The results in Section 5.3 were published in [17].

5.3.1 SNR 3:1 training data

Learning parameters are now selected to optimize performance in the SNR 0.3:1 regime for this comparative study, so the performance in the SNR 3:1 case will

not be as high as the best performance previously seen in Sections 5.1 and 5.2. In the case of SNR 3:1 training data, for both Flat CP and Chirped CP cases, the best accuracy in classifying SNR 0.3:1 test data was obtained by simple learned dictionaries with K=256 elements, at C=50 learning iterations (L_{train}=L_{class}=45). For each of the two data cases, a decision tree classifier is trained using the STFT coefficients of respective SNR 3:1 training data, as previously detailed in [17]. Five longer test timeseries (each 10 million samples long, resulting in 35102 labeled data windows), with different white noise random seeds, are classified for every SNR regime and both types of data.

Figure 5.13 shows resulting accuracy boxplots for learned dictionaries with K=256 elements, as well as resulting median accuracy over the five test data for the STFT classifier (green stars). The left side of Figure 5.13 shows results for Flat CP data case, and the right side for Chirped CP data case. In both cases, classification is made on test data from all three SNR regimes (3:1, 1:1, and 0.3:1). The 10 pairs of ON/OFF learned dictionaries are used on the 5 test timeseries, resulting in 50 classification accuracy estimates. The boxplots show therefore a more representative range of accuracies, but account for both variation in the noise seed for the test data, as well as variation in the noise used to seed the set of learned dictionaries. Figure 5.13 shows that learned dictionaries trained on high SNR data are more robust to changes in the SNR of the test data than the STFT. It appears that accuracy performance for the STFT-classifier is very high for SNR 3:1 test data (as noted by the green stars), but degrades severely as the SNR worsens (again compare the low

value as denoted by the green stars in the 1:1 and 0.3:1 cases). In contrast, the learned dictionaries of the size and learning parameters selected here do not reach as high an accuracy performance on the SNR 3:1 test data, as expected, since the choice was to optimize for SNR 0.3:1. However, they degrade in performance more gracefully as the SNR worsens compared to the STFT-classifier, and lead to higher median performance on SNR 1:1 and SNR 0.3:1 data. A better interpretation of the results in Figure 5.13 can be made after the behavior is similarly explored in different training conditions, that is, at the end of Section 5.3.2.

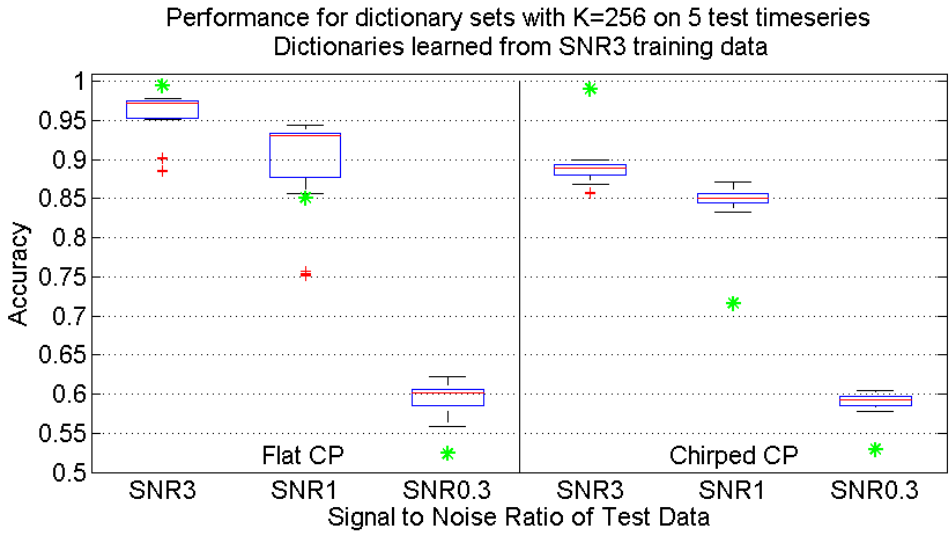


Figure 5.13: Classification accuracy boxplots for the two clutter scenarios, left side Flat CP data case, and the right side Chirped CP data case, using MR classifier with learned dictionaries of K=256 (boxplots), and decision tree classifier with STFT (green stars). SNR 3:1 training data used in all cases.

5.3.2 SNR 1:1 training data

In this case, SNR 1:1 training data is used to similarly learn dictionaries, and train a STFT-based decision tree classifier. As in Section 5.3.1, one dictionary size is selected for this study such that performance is optimal in the SNR 0.3:1 test case for both data types. This choice reflects the ongoing priority to make classification practical in noisy environments. When using SNR 1:1 training data, the best classification performance is obtained with simple K=64 dictionaries (L_{train}=L_{class}=15) at C=31 learning iterations for the Flat CP case, and C=23 for the Chirped CP case. Figure 5.14 shows the corresponding accuracy when classifying 5 test timeseries in each SNR regime with learned dictionaries (boxplots), as well as the median accuracy obtained with the STFT classifier (green stars). The left side of Figure 5.14 shows results for Flat CP data case, and the right side for Chirped CP data case.

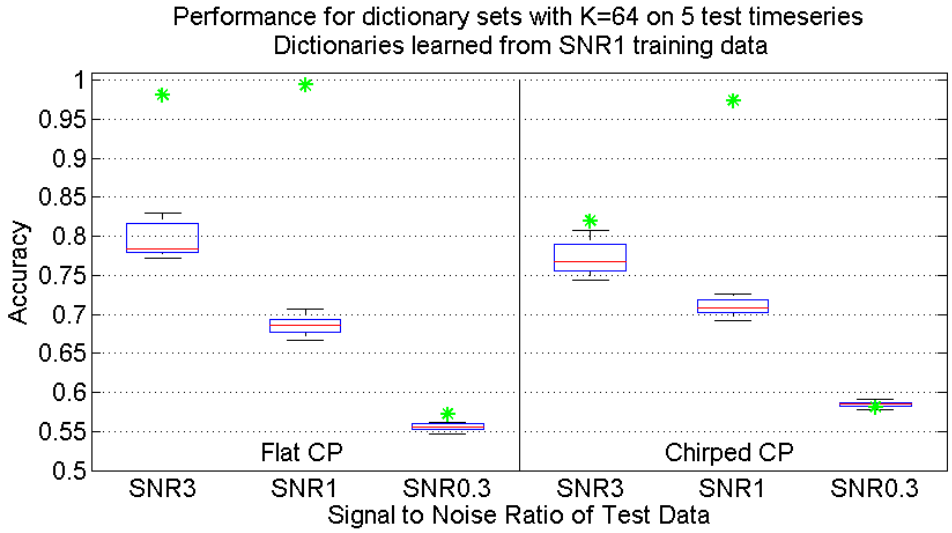


Figure 5.14: Classification accuracy boxplots for the two clutter scenarios, left side Flat CP data case, and the right side Chirped CP data case, using MR classifier with learned dictionaries (boxplots), and decision tree classifier with STFT (green stars). SNR 1:1 training data used in all cases.

Figures 5.13 and 5.14 illustrate how learned dictionary-based classification responds differently to changes in SNR compared to STFT-based classification, and this is where the defining difference in behavior between the two methods is observed. For the learned dictionaries, effective training requires high SNR data. When the training data come from the SNR 3:1 set (boxplots in Figure 5.13), the classification accuracy is better in all cases than when the training data come from the SNR 1:1 set (boxplots in Figure 5.14). In both cases, the results are better for the data containing a simpler (i.e., flat) clutter pulse, and in both cases the accuracy follows the changes in test data SNR (e.g., it decreases as the SNR of the test data decreases). That is, in all cases, *the performance is dominated by how easy or difficult it is to distinguish the target from the background, rather than by how well the characteristics of the test data match the characteristics of the training data.* This is particularly noticeable in Figure 5.14, where the training data has an SNR of 1:1. Here the learned dictionaries perform better on the test data with SNR 3:1, in which the target is more distinguishable from the background, than they do on the test data drawn from the same distribution as the training data (SNR of 1:1).

In contrast, the green stars of Figures 5.13 and 5.14 show that the STFT-based classifier performs better than the MR with learned dictionaries when the test data come from the same distribution as the training data. As with the learned dictionaries, the STFT results are better for the data containing a simpler (i.e., flat) clutter pulse. However, the accuracy does not always degrade as the SNR of the test data decreases. Rather, *the accuracy degrades as a function of the difference between the test data*

and the training data. Again, this is most noticeable Figure 5.14, where the training data has an SNR of 1:1. The STFT performs better on the test data with an SNR of 1:1 than on the “higher quality" test data with an SNR of 3:1.

To summarize, across the different SNR regimes, the learned dictionaries perform best in cases where the training and test data have different background characteristics (i.e., 3:1 training with 1:1 or 0.3:1 test in Figure 5.13). In contrast, when the training and test data have similar characteristics (3:1 training with 3:1 test and 1:1 training with 1:1 test), the STFT shows better performance. A possible explanation for this behavior is that the STFT-classifier only captures coefficient amplitudes, and so classification decisions are made on the basis of amplitude alone. Learned dictionaries, on the other hand, can capture amplitudes as well as other characteristics of the data when the training data have a strong target signal, and so this richer representation of the features can allow for more robust classification when assumptions about the distribution of amplitudes fail or cannot be ascertained.

In conclusion, learned dictionaries are more robust to changes in the SNR of the test data than the STFT, and their performance depends upon how easy or difficult it is to distinguish the target from the background. Learned dictionary accuracy is strongly impacted by changes in the SNR of the training data. Indeed, the dependence is so strong that better accuracy is obtained by training on high SNR data, even if the test data have low SNR. In contrast, the STFT performance depends upon how well the characteristics of the test data match those of the training data. This dependence is so strong that better accuracy is obtained when test data SNR matches the training

data SNR than when the SNR of the test data increases. *Learned dictionaries could therefore be more useful for cases where training and test data belong to* different *(unknown) SNR regimes, while STFT should be employed when the* same *SNR regime is expected.* For both methods the results are better for the data containing the simpler (flat) clutter pulse, as expected.

Even though only a particular combination of dictionary parameters is used here, previous sections showed a range of parameter values which result in dictionaries with similar relative behavior in classification, for some different dictionary sizes (from 0.125 to 0.5 times undercomplete). Based on those results, one can extrapolate that classification response to changes in SNR as described in this section follows the same pattern (i.e., higher accuracy for louder target, lower accuracy for quieter target), for a range of undercomplete dictionary sizes and a range of learning parameters. Also, the particular dictionary selection for this section was made for peak performance in the SNR 0.3:1 regime. Previous sections demonstrated better accuracy can be obtained on SNR 3:1 and 1:1 test data with different dictionary choices, and that performance is much closer to the STFT performance for the higher SNR regimes. The following section will present a new ensemble classification method that can lead to more stable (i.e., with less variation) accuracy performance, and would permit simultaneous optimization of classification across different SNR regimes.

5.4 Stochastic classification: Minimum Residual Ensemble Classifier (MREC)

Even though a final, learned dictionary and its performance are deterministic, the dictionary learning process is stochastic, and leads to the variations we have seen in all previous boxplots. A novel learned dictionary voting system is now introduced, called a *Minimum Residual Ensemble Classifier* (MREC). This ensemble classifier was developed by the author to address specific needs (outlined below), and its benefits are demonstrated in this section. Similar performance on a different dataset was separately shown by the author in [16].

Accuracy improvements can be achieved by polling multiple minimum residual classifiers using distinct pairs of ON/OFF dictionaries, where each pair is learned from a different random initialization. Given multiple votes (one from each dictionary pair), a window is labeled "ON" if the majority of dictionary pairs return an "ON" vote (i.e., if it receives an ***ensemble*** "ON" vote).

The performance on the Chirped CP data set can be improved using this ensemble classifier. For example, consider the case of a very undercomplete dictionary with K=64 elements and L_{train}=8, L_{class}=8, and C=33 learning iterations (Figure 5.4). Here the median accuracy on SNR 3:1 test data is only 0.875 and it exhibits wide variations (±0.75). On the other hand, in Section 5.3, the K=64 dictionary (used with slightly different learning parameters) gave the best performance on SNR 0.3:1 data. To make classification practical in noisy

environments, a good classification scheme with learned dictionaries would have optimal accuracy not just for a single SNR regime, but for a wider range of SNR regimes. The MREC directly addresses this practical need, and its effectiveness is demonstrated below.

One hundred dictionary pairs with K=64 and fixed learning parameters were learned from SNR 3:1 Chirped CP training data. The MREC is evaluated by gradually increasing the number of voting dictionary pairs and looking at the variance across a random subset of all possible combinations. Since in the 1 voter case there are only 100 combinations, the subset size considered is 100 voting groups. The cases of 1 through 9 voters in a group are specifically considered, where a voter is a particular ON/OFF dictionary pair, randomly selected from the full set of 100 possible voters without replacement. For each number of voters V, 100 groups with random combinations of the 100 dictionary pairs taken V at a time are selected. A group of V voters casts an ensemble vote in the manner described above, and the overall classification accuracy variance is captured in the boxplots of Figure 5.15. In every test scenario the ensemble classification performance increases with the number of voters, as expected. This improvement is characterized both by the increase in median accuracy (red lines), as well as the decrease in variation (i.e., height of boxplot) across a voting set of 100-choose-V. In fact, median classification accuracy improves by up to ~0.15 as the number of voters increases, and its variance decreases by a significant 92%. As the voting group size increases past V=7, the MREC accuracy appears to remain relatively similar in terms of variance and median value over the

random sets of 100 pairs. The best median accuracy for dictionaries with K=64 is now ~0.94±0.014, which is a much improved result compared to the simple MRC case.

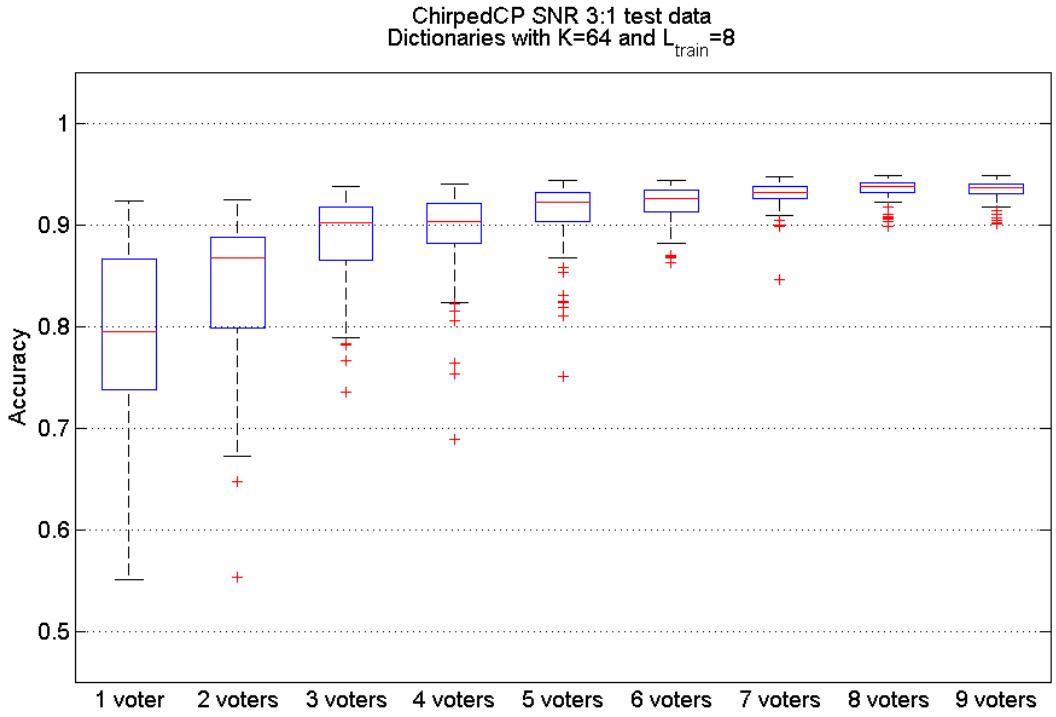


Figure 5.15: Ensemble classification accuracy (y-axis) as a factor of number of allowed voters (x-axis) for each test data set. A boxplot represents 100 selected combinations of V voters (100-choose-V, with V=1, 3, 5, 7, 9) out of the 100 learned dictionary pairs for dictionary size K=64 and L_{train}=8. Median classification accuracy (red lines) improves by 0.15 between 1 and 9 voters, and its variance (i.e. height of boxplot) decreases by a factor of ~12.5 (i.e., by 92%)!

In an actual implementation with, e.g., 9 voting dictionaries, the MREC is implemented in a parallel process, where each of the 9 dictionary pairs classifies input data simultaneously on different cores, with little added computational overhead due to communication time. The MREC allows for good classification with smaller dictionaries, which helps reduce computational overhead associated with learning a larger dictionary. In this parallel implementation, it is feasible to have the voters be multiple dictionaries of *different* sizes and/or optimized for different noise conditions,

leading to a multiscale learned dictionary analysis tool that can provide additional reliability and decision confidence.

5.5 Conclusion on classification with undercomplete learned dictionaries

One of the main hypotheses in this thesis is that perfect reconstruction is not necessary for high classification performance. This chapter supports this hypothesis in two ways. First, it shows that dictionaries with varying degrees of undercompleteness can give high classification performance for a range of parameter values, depending on the SNR case. That is, overcomplete dictionaries that might be necessary for high reconstruction performance are not necessary for high classification performance.

Secondly, this chapter demonstrates that a finer approximation (i.e., larger sparsity factor, L), which is synonymous with "better reconstruction," is at best not improving classification, or is at worst hindering classification, compared to a coarser approximation (i.e., smaller sparsity factor, L).

The up-front expense of using learned dictionaries can be quite large; for instance, the sensitivity analysis presented in this chapter took approximately 57 hours of run time on 8 Intel Xeon processors run in parallel at 2.67GHz. The learning cost for a single Hebbian dictionary is however very small, of the order of few minutes depending on the value of K, C, and the amount of training data. The advantage is that learned dictionaries for classification can be undercomplete, and the

classifier has a very simple implementation, leading to low computational overhead in the classification stage. Given the better robustness to SNR changes compared to a STFT-classifier, the learned dictionaries offer a decided advantage, and may be worth the up-front expense in return for more robust performance at test time.

Thirdly, it was shown that hybrid learned dictionaries can provide added robustness in classification across different SNR regimes, and they can be useful in applications where an extensive sensitivity analysis on the learning parameters may not be feasible. For undercomplete Hebbian dictionaries, the hybrid algorithm provides faster learning convergence and more stable asymptotic behavior with respect to classification accuracy.

Lastly, a new minimum residual ensemble classifier (MREC) was introduced and shown to significantly improve median accuracy and accuracy stability compared to the MRC. The MREC takes advantage of parallel computing to reduce computational time by querying dictionary pairs simultaneously, and can be implemented as a multiscale tool by using voting dictionaries of different sizes to enhance robustness to noise.

6. Classification with Undercomplete Dictionaries in Satellite Imagery using CoSA

Land cover classification in satellite imagery presents a different type of signal processing challenge. The classification technique employed for the synthetic RF target cannot be directly extended to this new problem for several reasons. Chief among these is the lack of verified correlation between image data and real features on the ground (so-called 'ground truth'), which precludes any direct supervised classification. Remote sensing techniques can analyze multispectral satellite data and perform coarse land cover classification, e.g., based on coefficient binning in the Normalized Difference Vegetative Index (NDVI) image. Another approach would be applying genetic algorithms, such as Genie [123], to extract a particular class of interest in a supervised manner. Such methods rely heavily on domain expertise and usually require human input.

Techniques for automated feature extraction and classification are of current interest in the areas of climate change and Land Use/Land Cover classification using satellite image data [123-128]. This chapter builds on the knowledge gained in Chapters 4 and 5 to present a technical solution for automatic classification of land cover in multispectral satellite imagery of the Arctic using sparse representations in undercomplete Hebbian learned dictionaries: ***clustering on sparse approximations (CoSA)***. The method is demonstrated using DigitalGlobe Worldview-2 visible/near

infrared high spatial resolution imagery. The Hebbian learning rule detailed in Chapter 4 is used to build dictionaries that are adapted to the data. Sparse image representations of pixel patches over the learned dictionaries are used to perform unsupervised *k*-means clustering into land-cover categories. This approach combines spectral and spatial textural characteristics to detect geologic, vegetative, and hydrologic features. Performance is evaluated mostly qualitatively and the purpose of this chapter is to demonstrate how the methods introduced in Chapters 4 and 5 can be adapted to an entirely different application. An in-depth analysis is ongoing and will be presented in forthcoming publications. Results suggest that neuroscience-based models are a promising approach to practical pattern recognition problems in remote sensing, even for datasets using spectral bands not found in natural visual systems. Using undercomplete dictionaries provides dimensionality reduction, which is desirable in high data rate applications. The following sections expand upon each of the specific steps, beginning with a summary of climate observations that frame the problem.

6.1 Introduction

Recent work in the area of climate change monitoring has indicated that air temperatures have been rising in both Alaska [129, 130], and the western Canadian Arctic [129, 131-133] over the last few decades. Global climate models suggest that the Arctic will continue to warm more rapidly than more southerly locations [129], therefore significant attention has been given to this region by climatologists. The

Arctic biome (both terrestrial and aquatic) is responding to this warming, and the effects are multiple and interconnected. In the terrestrial biome, the thawing of the permafrost (soil at or below the freezing point of water 0°C (32°F) for two or more years) is directly correlated to the vegetative cover. Specifically, warming alters transitional regions between upright and dwarf shrub tundra [134], and there has been evidence of increasing shrub cover on air photos [135], and of changes to vegetation indices derived from satellites [136-139]. Colonizing shrubs affect snowpack depth, although it is yet unclear at what magnitude and in what direction [133]. Along arctic coastlines and riverbanks, recent studies have attributed dramatic increases in the rates of shoreline erosion to global climate change and near-surface permafrost degradation [140, 141]. Across much of the arctic, the number of lakes and their sizes have also been changing as a result of permafrost degradation, altering surface water dynamics and causing possible release of soil organic carbon (SOC) [142, 143].

Currently, climate change experts primarily use various indices derived from the spectral bands, such as the Normalized Difference Vegetative Index (NDVI - a normalized pixel-level combination of two spectral bands), which is here calculated as the (NIR-RED)/(RED+NIR) bands. Other land cover classification approaches involve the use of state-of-the-art genetic algorithms, such as Genie [123, 125]. The NDVI is one of the most successful methods to simply and quickly identify vegetated areas and their "condition," and it remains the most well-known and used index to detect live green plant canopies in multispectral remote sensing data [144]. The rationale behind NDVI comes from a.) the chlorophyll in plant leaves strongly

absorbs visible light (from 0.4 to 0.7 μm) for use in photosynthesis, and b.) the cell structure of the leaves strongly reflects near-infrared light (from 0.7 to 1.1 μm). Therefore the NDVI is directly related to the photosynthetic capacity and energy absorption of plant canopies, and areas containing dense vegetation will tend to positive NDVI values (~0.3 to 0.8). By contrast, features such as clouds and snow tend to be rather bright in the red (as well as other visible wavelengths) and quite dark in the near-infrared, leading to negative NDVI values. Other targets, such as water bodies or soils, will both generate small positive NDVI values, or in the former case, sometime slightly negative NDVI values, and thus they are not distinguishable with confidence using NDVI. The downside is that domain expert input (e.g., a human worker) is required to properly decide the binning of NDVI coefficients, and the NDVI index is sensitive to several factors and not a strictly robust approach.

Genie is a feature extraction tool developed at Los Alamos National Laboratory for multispectral, hyperspectral, panchromatic, and multi-instrument fused imagery, but it too requires supervision in training. One of the main limitations is the difficulty in providing clean training data, i.e., only pixels that truly belong to the class of interest. This is easier to do when the classes are well separated spatially from other classes, such as water bodies from land, or golf courses from buildings [145], but much more difficult to do when the classes are intermingled.

Such approaches perform well on certain types of problems, such as distinguishing between water and land features (i.e., automatic lake detection), or specific vegetative analysis, but typically are not robust for multiple classes that come

from different categories. It is therefore of great importance to develop satellite imagery analysis techniques that would allow automatic classification of many land cover categories and be feasible for high resolution change detection in land cover. These techniques would provide the climate change community with more exact ways of monitoring changes and quantifying various effects. The work in this chapter motivates a classification approach based on undercomplete learned dictionaries that would address this need for multiclass detection.

6.2 MacKenzie watershed satellite data

DigitalGlobe's Worldview-2 satellite imagery [146], used in this thesis, is the highest resolution commercially available multispectral data at 1.84m spatial resolution. The WorldView-2 sensor provides eight multispectral bands: four standard wavelengths (red, green, blue, and near-infrared 1) and four new bands. Ordered from shorter to longer wavelength the list of bands is: coastal, blue, green, yellow, red, red-edge, near-infrared 1, and near-infrared 2.

The specific problem chosen to illustrate application of learned dictionaries is automatic land cover detection in the Trail Valley Creek region of the Mackenzie River watershed (Figure 6.1). Shown here is a 3-band image formed with the traditional red, green, and blue (RGB) bands. The true spatial extent of the region is approximately 6 km x 10 km. There are many features of interest present, primarily vegetative and geomorphic, but also aquatic; an ideal classification scheme would be able to simultaneously identify them all.

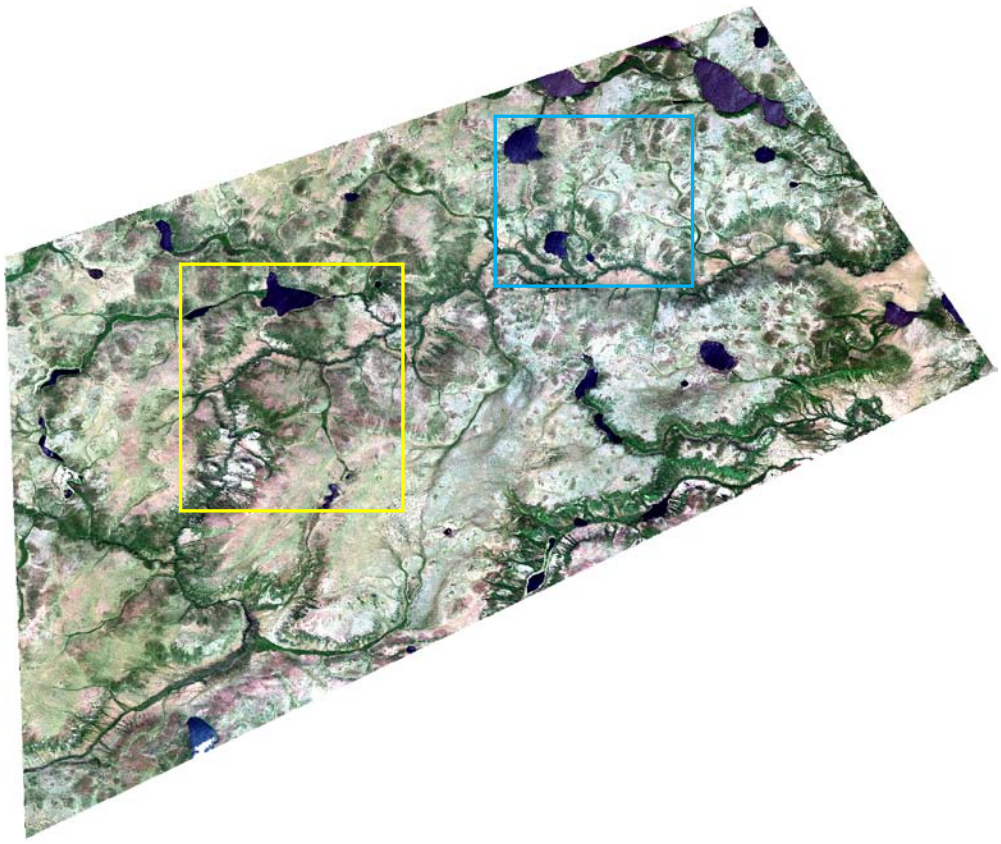


Figure 6.1: Trail Valley Creek watershed, east of the Mackenzie River, NW Canada (Worldview-2 satellite data). Full basin RGB image (approximately 6 km x 10 km spatial extent). There are many features of interest present, primarily vegetative and geomorphic, but also aquatic. The area delineated by the yellow rectangle is the control image used in the remainder of the chapter. The area delineated by the cyan rectangle is a validation image used for cluster analysis in Section 6.4.

A high-level classification of land cover in this region is given by Marsh [133] (Figure 6.2), who focused on using vegetation height for snowpack analysis. He performed a simple low-resolution classification using LiDAR data, and identified only four categories: tundra, low shrub, tall shrub, and trees. The approximate region shown in Figure 6.1 is marked with a red rectangle in Figure 6.2.

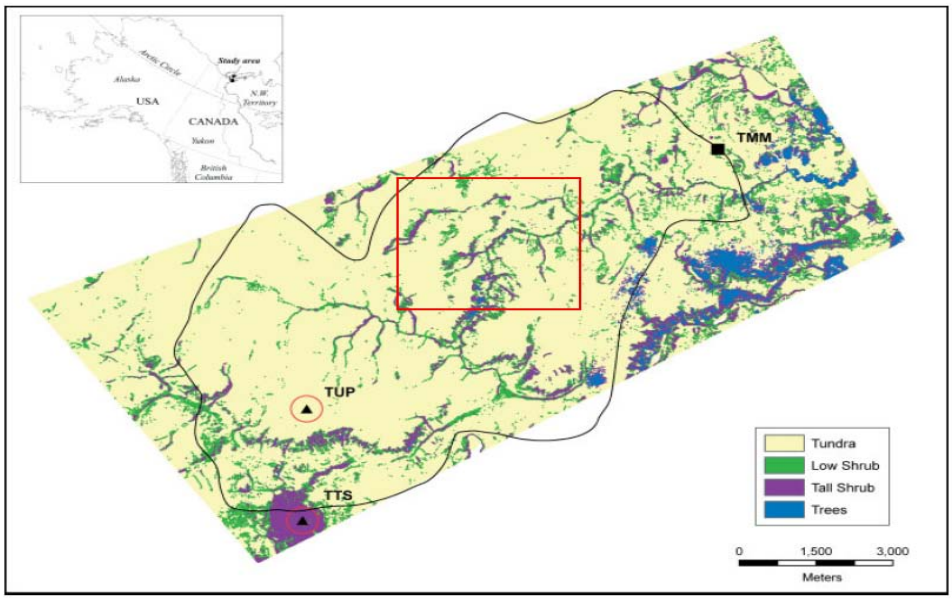


Figure 6.2: Trail Valley Creek watershed (with basin boundary shown by black line, and basin outlet to the east). Only four vegetation height classes were derived from LiDAR, with tundra defined as areas with vegetation <0.50m, low shrub defined as vegetation >0.50m and <1.25m, tall shrubs >1.25m and <3.0m, and trees >3.0m in height (Marsh et al. [133]).

It is obvious that for the level of detail present in Figure 6.1, more than four land cover classification labels are needed. For ease of visualization, the remainder of the chapter will show results on a control image (area delineated by yellow rectangle in Figure 6.1). A zoom of the control image is shown in Figure 6.3 in color infrared (i.e., bands 8, 6, 4: near-infrared 2, red-edge, yellow). This control region was selected because it includes the features of interest present in the entire image of Figure 6.1, such as polygonal ground (blue pixels in red rectangle), water (black pixels), lake drainage (green rectangle), various vegetation types (yellow pixels), bare soil (cyan pixels). Among the features mentioned, polygonal ground is specifically encountered in permafrost regions. Such frozen soil can be dry, or it can contain ice, which is usually found in large wedges forming a honeycomb of ice walls beneath the

soil surface. The top soil buckles and cracks above the ice wedges, causing *polygons* to form at the surface that can be anywhere from ~70 feet (~20 m) across to ~10 feet (~3 m) across.

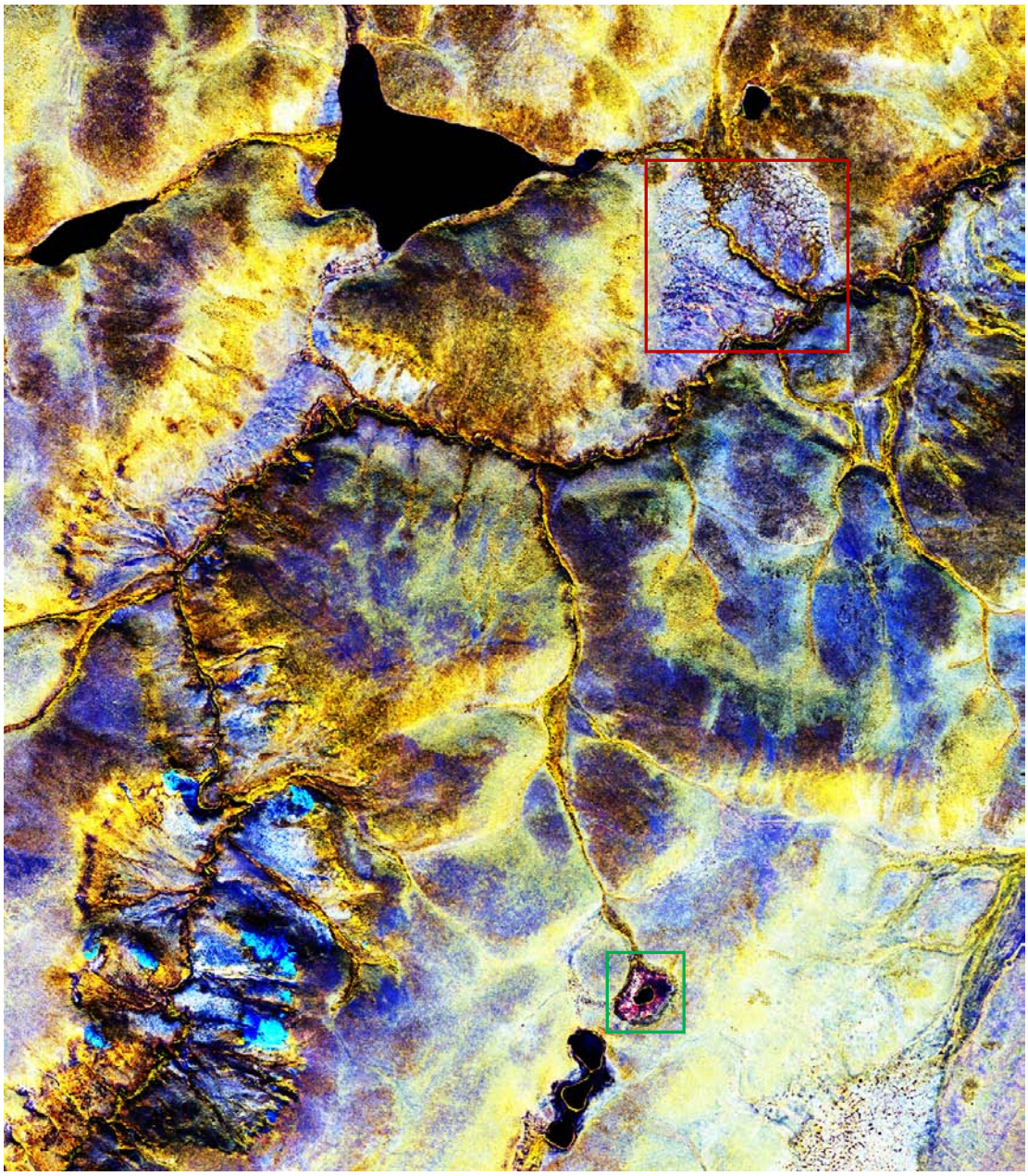


Figure 6.3: Control image zoom, shown in color infrared (i.e., bands 8, 6, 4: near-infrared 2, red-edge, yellow). Spatial extent is approximately 3km x 2.5km.

6.3 Multispectral undercomplete learned dictionaries

The CoSA technique presented in this chapter and in [19, 20] builds on the learned dictionary method explored in Chapters 4 and 5. A detailed sensitivity analysis on the parameters is not performed in this chapter, but rather the knowledge previously gained in classification of RF data is now extended to this new data type.

In terms of input dimensionality, the 1D time window used in the RF analysis of previous chapters is now replaced by a patch of p x p x 8 pixels, that is, the satellite imagery is processed using square multispectral pixel patches. A pixel patch is reshaped as a 1D vector of overall length N, where N is the natural input dimensionality. Even though N might be large for satellite imagery, intuitively the "intrinsic" satellite data dimensionality may in fact be small, as it presents less category specific variability compared to, say, image data (e.g., there are few details visible from space for a patch of trees). The dictionary size, K, should depend upon this "intrinsic" data dimensionality, and this makes undercomplete learned dictionaries a prime candidate for this application. Satellite imagery is dominated by spatial and spectral texture (e.g., vegetation blend in a region), as opposed to individual features of a particular object class (e.g., leaf shape and bark color of a birch tree) in camera-images resulting in many very distinct pixel patches per object class. This is also true of urban satellite imagery, with added features from human-made structures. Most of the dimensionality in the satellite data is therefore primarily due to the number of categories present (i.e., all the different land cover classes), and not so much the fine details of each category. In the context of learned dictionaries, it

follows that lower values of K could in fact work very well for classification of satellite imagery.

Three different spatial resolutions of 7x7, 9x9, and 11x11 pixel patches are used to illustrate performance of undercomplete learned dictionaries. (Given the 1.84 m pixel resolution of the imagery, the chosen patch sizes map to physical square areas of length 12.8 m, 16.5 m, and 20.2 m, respectively.) These spatial resolutions result in natural dimensionalities of N=392, 648, and 968, and for each resolution a separate dictionary is learned. Each of the three dictionaries of different spatial resolutions is chosen to have a constant size of K=300 elements, making all three dictionaries undercomplete by a factor of 0.8, 0.5, and 0.3, respectively. This also means that all the sparse decomposition vectors are 300-coefficient long, regardless of the spatial resolution. The dictionaries are now initialized by imprinting (i.e., seeding the elements with random image vectors), to help speed up learning convergence. The rest of the learning parameters for the dictionaries are chosen following similar rationale to that previously described. The decomposition sparsity factor, L_{train}, is chosen to be 20 for all three resolutions, resulting in 20/300= 0.067 sparsity indices (defined as the fraction of the sparse approximation coefficients which are not zero). This value was determined based on a parametric sensitivity analysis (similar to that of Chapters 4 and 5) that is not included here due to space limitations. The learning iterations in this case stop whenever learning convergence is achieved (i.e., when the dictionary elements stop changing).

Millions of overlapping 8-band pixel patches are randomly extracted from the control region and used to learn spatial-textural undercomplete dictionaries with an on-line batch Hebbian algorithm with multiple learning iterations, as detailed in Chapter 4. Here the number of training vectors is larger by four orders of magnitude compared to the dictionary size, and faster learning convergence was observed, as expected. Recall that a batch algorithm means a single learning update for dictionary element φ_k is calculated not from a single input vector, but rather from a small batch of input vectors (in this case 10) as shown in equation (4.5).

Figure 6.4 shows quilts of the three 300-element dictionaries shown in RGB, where a dictionary quilt is a visualization method explained below. Each of the three dictionaries has different spatial resolution, which results in a different element length N. Every dictionary element of length N is in fact also a multispectral pixel patch and, when reshaped in the p x p x 8 format, it is possible to show an image of the element using only the corresponding RGB bands. Dictionary quilts are obtained by stacking the RGB images of all the elements in matrix form (those shown in Figure 6.4 are 30x10 elements). Each of the small squares in a quilt represents a dictionary element.

The quilts are a qualitative view of the dictionaries and provide insight into what specific land features are learned by each element. Upon visual inspection, almost all the elements exhibit texture (i.e., variability in pixel intensity), and many contain oriented edges similar to those in [13]. The author notes that some of the elements that appear more uniform in RGB exhibit more texture in other spectral bands; that is, some dictionary elements better capture the variability in other spectral

bands rather than in RGB. It also would appear that the dictionary with 11x11 spatial resolution has more “green” elements compared to the other two cases, a characteristic which is discussed qualitatively in Section 6.5.

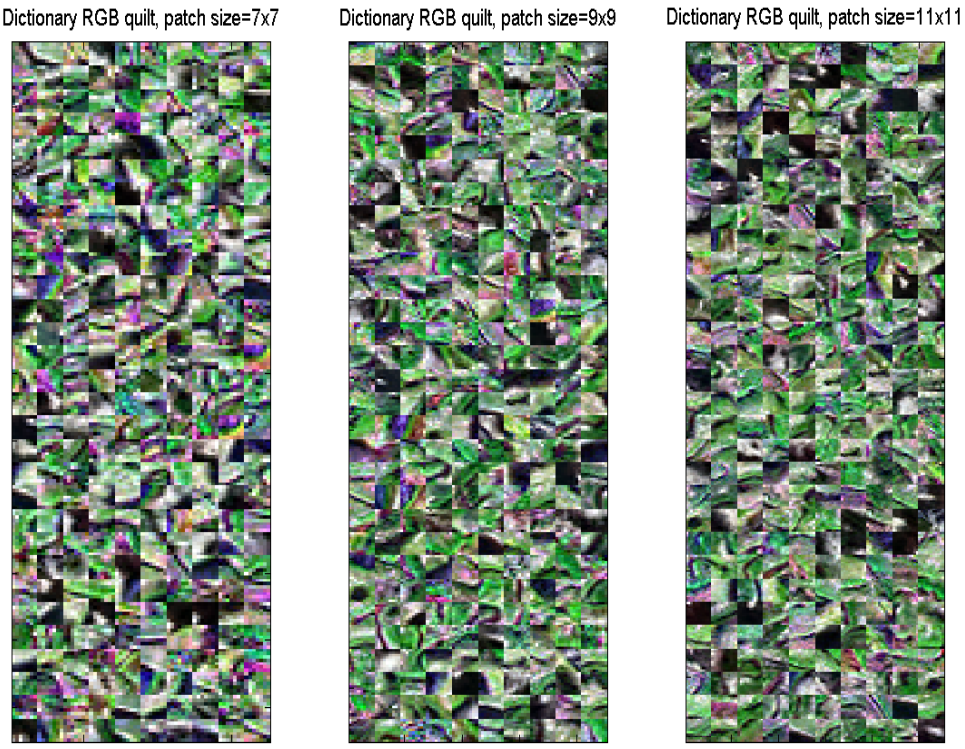


Figure 6.4: Quilts of dictionary elements (showing RGB channels of the 8-band element) for pixel patches of spatial extent 7x7 (left), 9x9 (center), and 11x11 (right). Each of the small squares in a quilt represents a dictionary element. All dictionaries have K=300 elements and are learned from the same training set. Visual inspection reveals dictionary elements have similar types of features across the three spatial resolutions explored.

6.4 Clustering of Sparse Approximations (CoSA)

To automatically classify land cover categories in an unsupervised fashion, the MRC of previous chapters can no longer be employed, as it requires labeled training data (i.e., ground truth). Instead, a *k*-means clustering algorithm [21, 44] is used on the sparse representations of the image patches, which in turn were found using the learned dictionaries of Figure 6.4.

The *k*-means algorithm is a well established data mining technique and aims to partition a set of unlabeled inputs (i.e., the sparse approximation vectors) into *k* clusters, where each input belongs to the cluster with the nearest center (in a Euclidian sense), and the value of *k* is user defined. The problem is NP-hard, but there are efficient heuristic algorithms that are commonly employed and converge fast to a local optimum. The *k*-means clustering seeks to find clusters in such a way that the total within-cluster variance (i.e., variance of distances to the cluster center) is minimized. Given an initial set of *k* centers (usually a random selection of *k* inputs), every algorithm iteration alternates between two steps:

- Assign each input to the closest cluster center (in this work the distance is considered to be Euclidian), i.e., *form the clusters;*
- For each cluster, calculate the new cluster center as the mean (centroid) of the observations in that cluster, i.e., *center the clusters.*

The algorithm is deemed to have converged when the assignment of inputs to clusters no longer changes. There are several methods to dynamically adjust the number of clusters, *k*, to fit the training data better [21, 44], such as splitting (i.e., dividing larger clusters into partitions), or pruning (i.e., eliminating empty clusters). In this demonstration the number of cluster centers is kept fixed.

For the particular problem in this chapter, the amount of data available is too large to be used in its entirety in finding the clusters centers. The solution is to use a subset of the data for determining (i.e., training) the clusters, and a separate validation

data subset to verify (i.e., test) the clusters. The control image is used as the training set, and the validation set is the region delineated by the cyan rectangle in Figure 6.1.

6.4.1 Cluster training

The clusters are trained from the sparse representations of pixel patches. For each of the three spatial resolutions, at every clustering iteration, overlapping pixel patches are randomly extracted from the control image in batches of fifteen thousand patches, reshaped as 1D vectors, and then decomposed over the respective learned dictionary using matching pursuit. The classification sparsity factor is selected to be equal to the learning sparsity factor, that is, L_{class}=20. The resulting sparse coefficient vectors are used to iteratively find cluster centers using the *k*-means algorithm.

One important question is determining the number of clusters necessary for good classification, from a domain expert point of view. A range of 4 up to 30 clusters is considered for each of the three spatial resolutions. Also, the *k*-means clustering algorithm is sensitive to initial conditions, and so the clustering should be checked for convergence across multiple initializations of the *k* centers. In this work, 15 different cluster initializations are considered for each number of centers. Figure 6.5 shows boxplots of median distances from the training vectors (i.e., sparse approximations of pixel patches in the control image) to their assigned cluster centers.

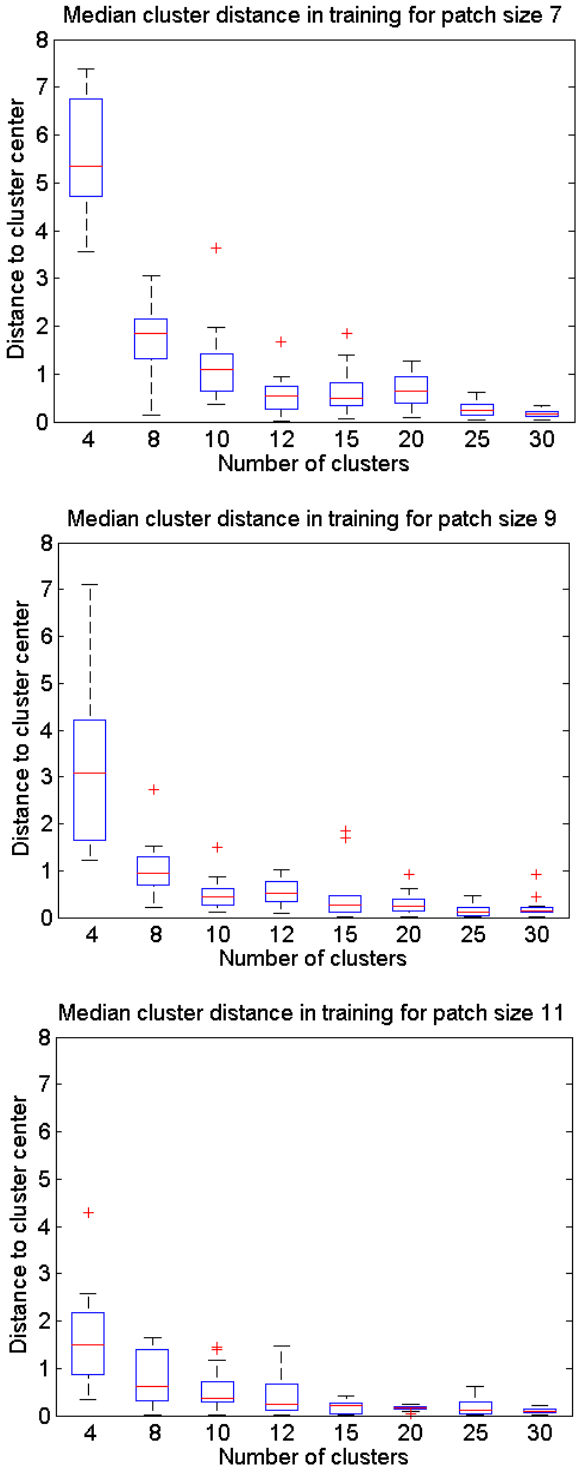


Figure 6.5: Training median clustering distance for 7x7 (top), 9x9 (middle), and 11x11 (bottom) spatial resolutions. Cluster distance and variance decrease as the number of clusters increases.

The boxplots summarize median distances across the 15 different sets of cluster centers for the training image, as a function of the total number of clusters, for the 7x7 case (top), 9x9 (middle), and 11x11 (bottom). As expected, median cluster distance for in-sample (i.e., training) data decreases as the number of clusters increases, and also the variance (width of the boxplots) of the median distance across the 15 different initializations decreases. In a direct comparison between the three spatial resolutions, the median cluster distance in training also decreases as the spatial resolution increases (Figure 6.6). Figures 6.5 and 6.6 show that, for training data, the performance remains relatively similar as the number of clusters increases past 15, especially for the 9x9 and 11x11 cases. This could be an indication that for this particular data set 15 is a good minimum value for the number of clusters.

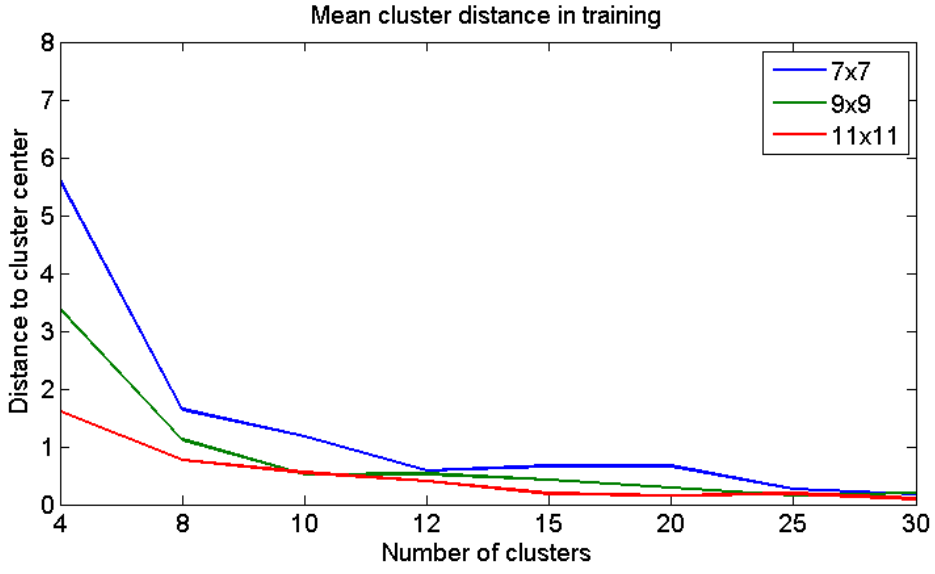


Figure 6.6: Mean training cluster distance for all 3 spatial resolutions. The mean cluster distance in training decreases as the spatial resolution increases.

6.4.2 Cluster testing

The validation image is now used to test how well the clusters obtained in Section 6.4.1 perform on a different, unseen data set. One way to test is to look at mean distances to cluster centers for each of the test patches and evaluate whether the clustering is still keeping these distances similarly small. At each spatial resolution, the validation image is processed using overlapping pixel patches (with 1-pixel step size), each reshaped as a 1D vector, and then decomposed over the respective dictionary using matching pursuit with L_{class} sparsity factor. For each of the resulting sparse coefficient vectors, the closest cluster center is found and the respective Euclidian distance is recorded. Figure 6.7 shows resulting distances from the test vectors to their assigned cluster centers. The boxplots summarize median distances across the 15 different set of cluster centers, as a function of the total number of clusters, for the 7x7 case (top), 9x9 (middle), and 11x11 (bottom). Note that the *y*-axis scale is only a half of the scale used in Figure 6.5 for the training median distances. The clustering of observations in the test data appears to be working well and the within-cluster distances are remaining reasonably small.

The results in Section 6.4 show that a minimum of 15 clusters is necessary for good partitioning of the sparse representations for the data considered here. An upper bound for the number of clusters could be identified if the distances to the cluster centers begin increasing as the number of centers increases, but there is no clear indication of divergent behavior within the number of clusters investigated (i.e., within 30 cluster centers).

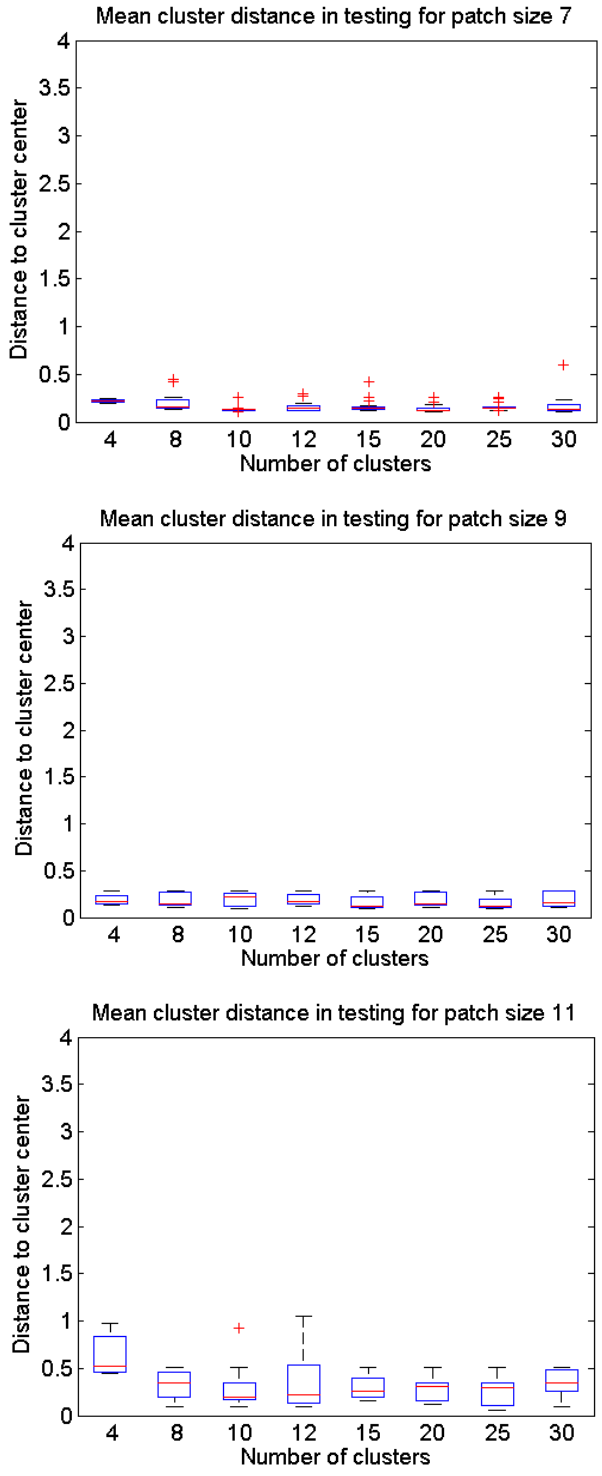


Figure 6.7: Testing mean clustering distance for 7x7 (top), 9x9 (middle), and 11x11 (bottom) spatial resolutions. Mean cluster distance and variance decrease as the number of clusters increases.

6.5 Land cover categories

Lack of pixel-level, verified, ground truth in satellite imagery prevents use of established quantitative classification performance metrics. Other approaches to estimating ground truth labels involve manipulations of spectral bands into index images, e.g., NDVI index, and supervised assignment of values to image categories. The NDVI index image using bands 7 (NIR) and 5 (red) for the control image in Figure 6.3 is shown in Figure 6.8. Purple and blues indicate low vegetation areas, reds represent high vegetative indices. The tundra exhibits a wide variation in greenness, depending on its aspect (i.e., north versus south facing slopes). Given that this particular image was taken in June, which is the Arctic spring and therefore early in the growth season, the south-facing slopes have more vegetation (red). Trees mapped along the stream corridors standout as red in the NDVI map. As previously mentioned, the NDVI does not easily discern between moisture rich features such as bare soil and water, and the area delineated by the black rectangle show this limitation (the purple areas there are in fact bare soils, not water).

The performance of the CoSA technique can now be qualitatively evaluated. Referring back to the four categories of land cover in Figure 6.2 (tundra, low and tall shrubs, and trees), the hope is that at least those categories could be reliably identified using CoSA. Compared to the NDVI index in Figure 6.8, the expectation is that CoSA would segment the control image in similar categories, would preserve the texture, especially in areas of transitional vegetation (e.g., where greens and yellows intermingle), and would resolve confusion between categories (e.g., soils and water).

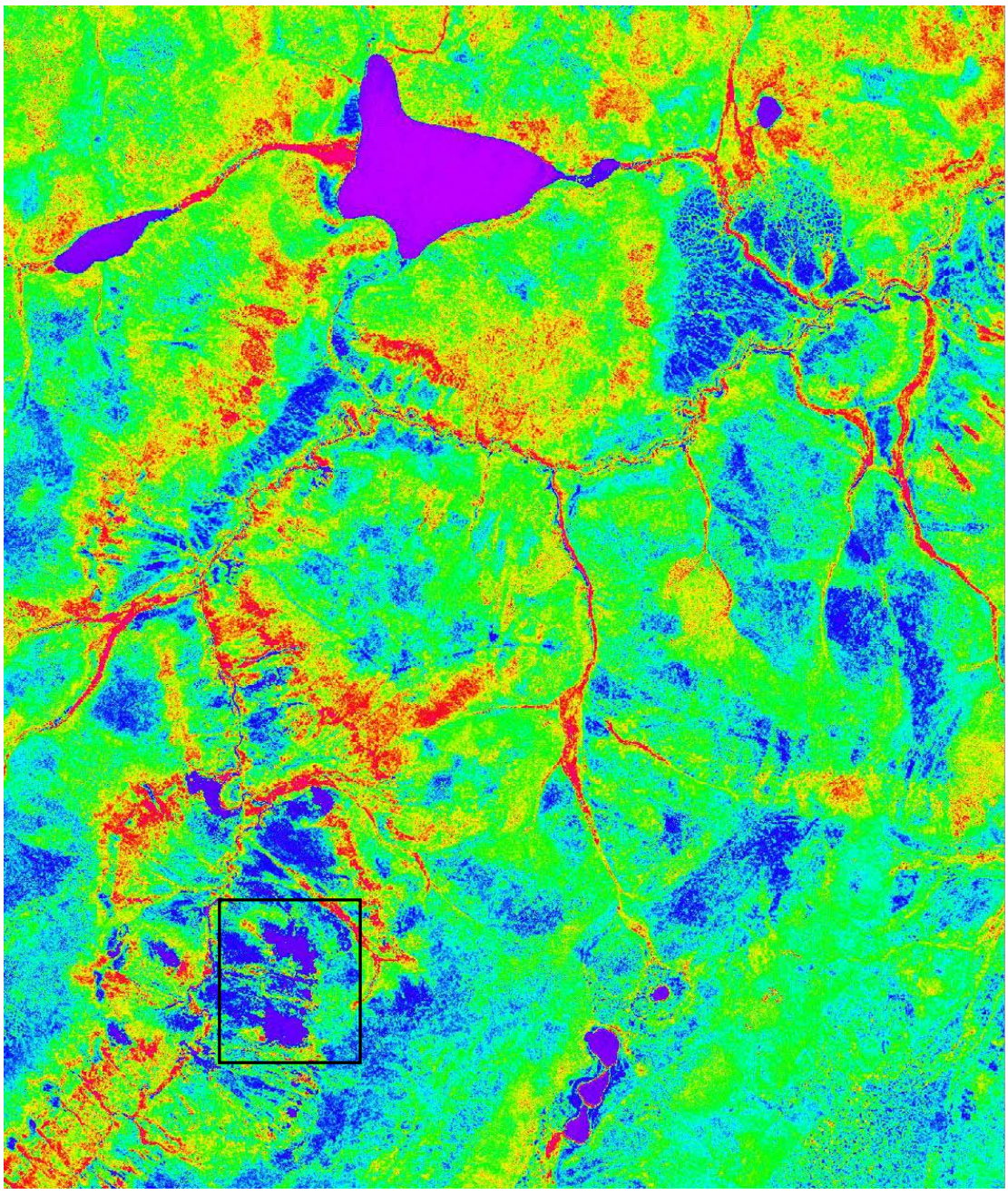


Figure 6.8: NDVI index map. Purple and blues indicate low vegetation areas, reds represent high vegetative indices. The tundra exhibits a wide variation in greenness, depending on its aspect (i.e., north versus south facing slopes). Trees mapped along the stream corridors stand out as being red in the NDVI map.

As in Section 6.4, all the overlapping pixel patches in the control image are sparsely represented using the learned dictionary, and the resulting sparse approximation vectors are assigned the labels of the nearest cluster centers. Example classification labels for each resolution for a single number of cluster centers are shown in Figures 6.9-6.11. Here each color in the image represents a particular cluster label. A quantitative way to select the optimal number of clusters at each resolution is by using the mean cluster distance analysis in Section 6.4. Those results showed that a minimum of 15 clusters was necessary for good partitioning of the data, but there was no clear maximum number of clusters identified. The final selection of the number of clusters is based on analysis of CoSA results with more than 15 clusters by a domain expert, Dr. Joel Rowland (LANL). He considered whether the clustering gives meaningful separations, and whether it is better than NDVI binning.

At 7x7 spatial resolution (12.8 m), the best CoSA result is obtained with 20 clusters (Figure 6.9). CoSA is clearly behaving more similarly to the NDVI analysis, in that it detects a wide variation in greenness of the tundra and presents meaningful texture. The water and the bare soil (the dark red areas delineated by the black rectangle) are still lumped together as in Figure 6.8, which makes sense if CoSA uses the spectral information similar to NDVI (i.e., both water and soil have very low vegetation related signals). However, another aspect stands out, not as easily explained: water bodies and tree-dense regions along the stream channels are also lumped together in Figure 6.9 (dark red label), whereas in the NDVI they are on opposite ends of the spectrum.

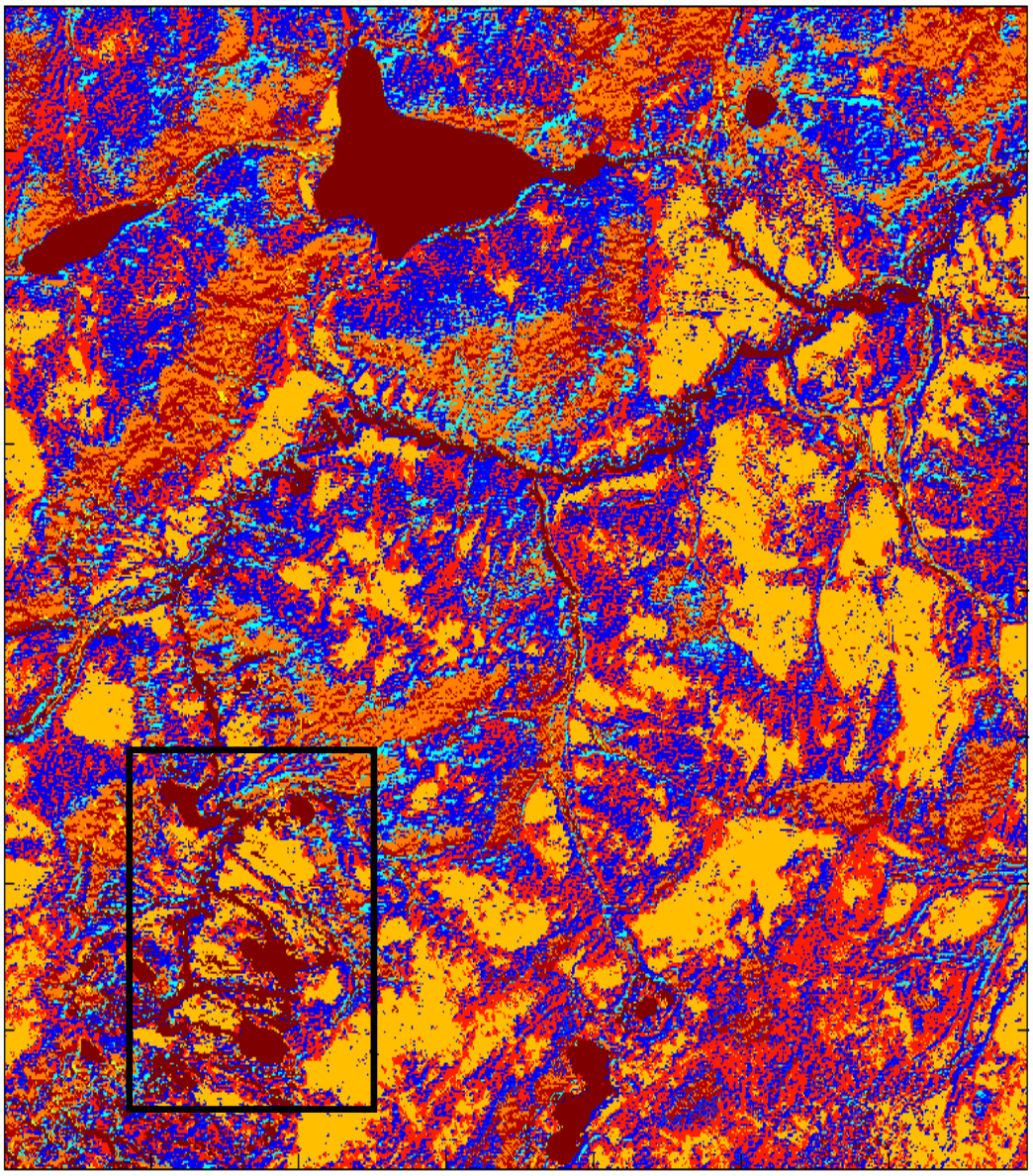


Figure 6.9: Category labels for 7x7 pixel patch. Analysis with 20 clusters detects a mix of vegetations types. Large water bodies and the tree dense regions along the stream channel are being clustered together, whereas in the NDVI these areas are on the opposite end of the spectrum. The clustering of the water with the bare soil (the white areas in the SW) is consistent with both having very low vegetative related signals (rectangular markup).

At 9x9 spatial resolution (16.5m), the most interesting CoSA result is obtained with 30 clusters (Figure 6.10). The water bodies (blue label) are now in a separate cluster from the soils (cyan class), and they are also in a different class from the dense tree regions along river streams (darker blue label). In Figure 6.9 the clustering is predominantly corresponding to vegetation distributions, but in Figure 6.10 the non-vegetation aspects of the landscape emerge. There is still abundant texture in the vegetation labels (bright reds and yellows), but there is also a lot of apparent texture due to geomorphic features. The area delineated by the black rectangle east and south east of the big lake is a good example of these geomorphic clusters. In the eastern portion (the yellow with the red and orange), CoSA could be detecting the round polygonal ground, and in the aqua/cyan region further west of there with the maroon linear clusters, CoSA might be picking up rills (i.e., poorly developed channels) that cause somewhat regular variations in the surface topography.

When the spatial resolution increases to 11x11 (20.2 m), at 30 clusters (Figure 6.11) CoSA is showing a lot more coherence in the cluster blocks compared to Figure 6.10. Water bodies, soils, and trees along stream channels are each in different clusters, but the non-vegetative features are not detected as well in this case as they were in the 9x9 case. There is good clustering of the lake shoreline vegetation (compare to Figures 6.9-6.10). The vegetative clusters in Figure 6.11 also appear to retain transitional vegetation information in the shrub and grass areas (e.g., light yellow transitioning into lime green), which, if proven valid, could be useful for detection of colonizing shrubs.

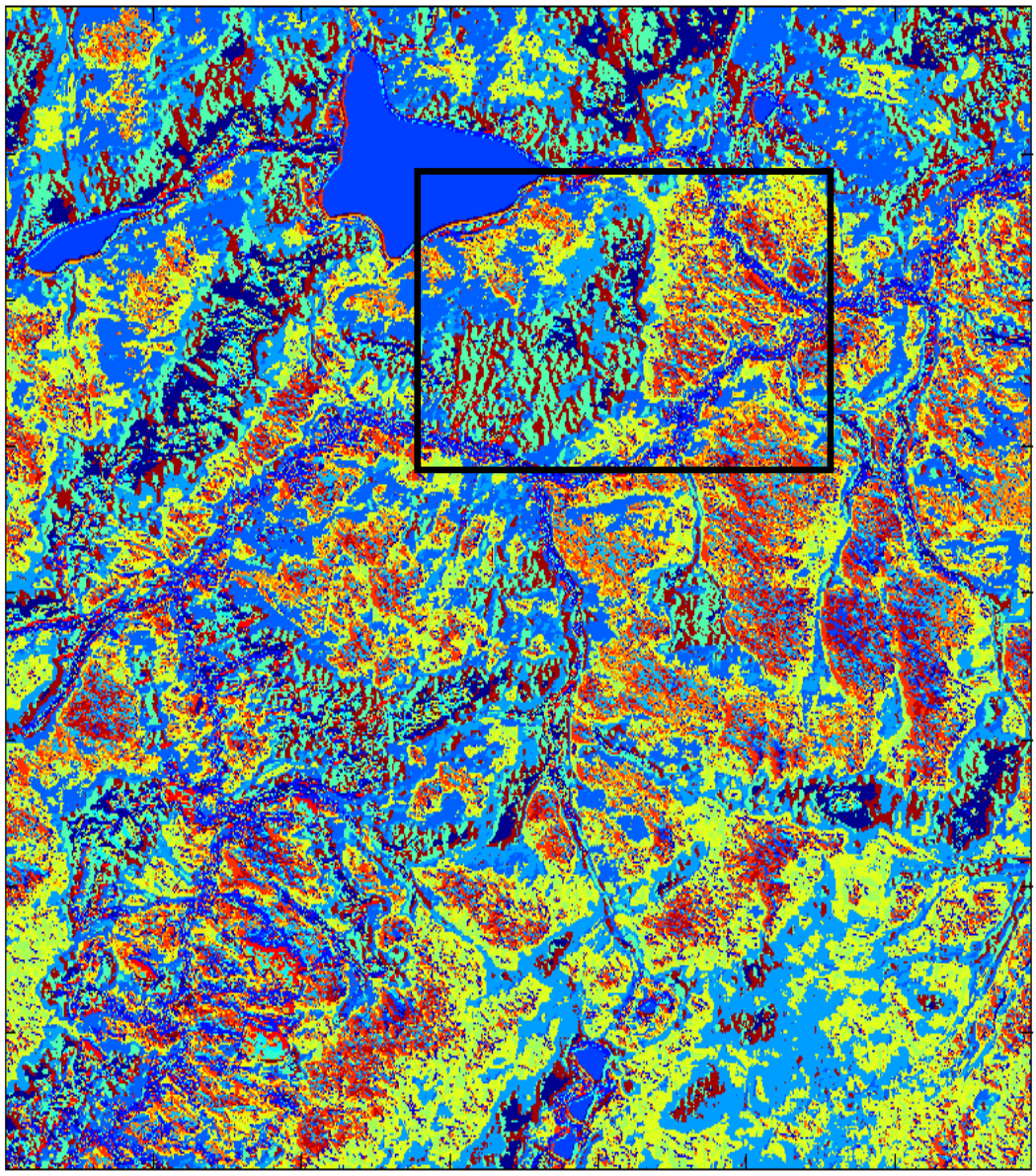


Figure 6.10: Category labels for 9x9 pixel patch. Analysis using 30 clusters allows the non-vegetation aspects of the landscape emerge. Looking east and south east of the big lake (rectangular markup) abundant texture is observed indicating round polygonal ground and rills (poorly developed channels) that cause somewhat regular variations in the surface topography.

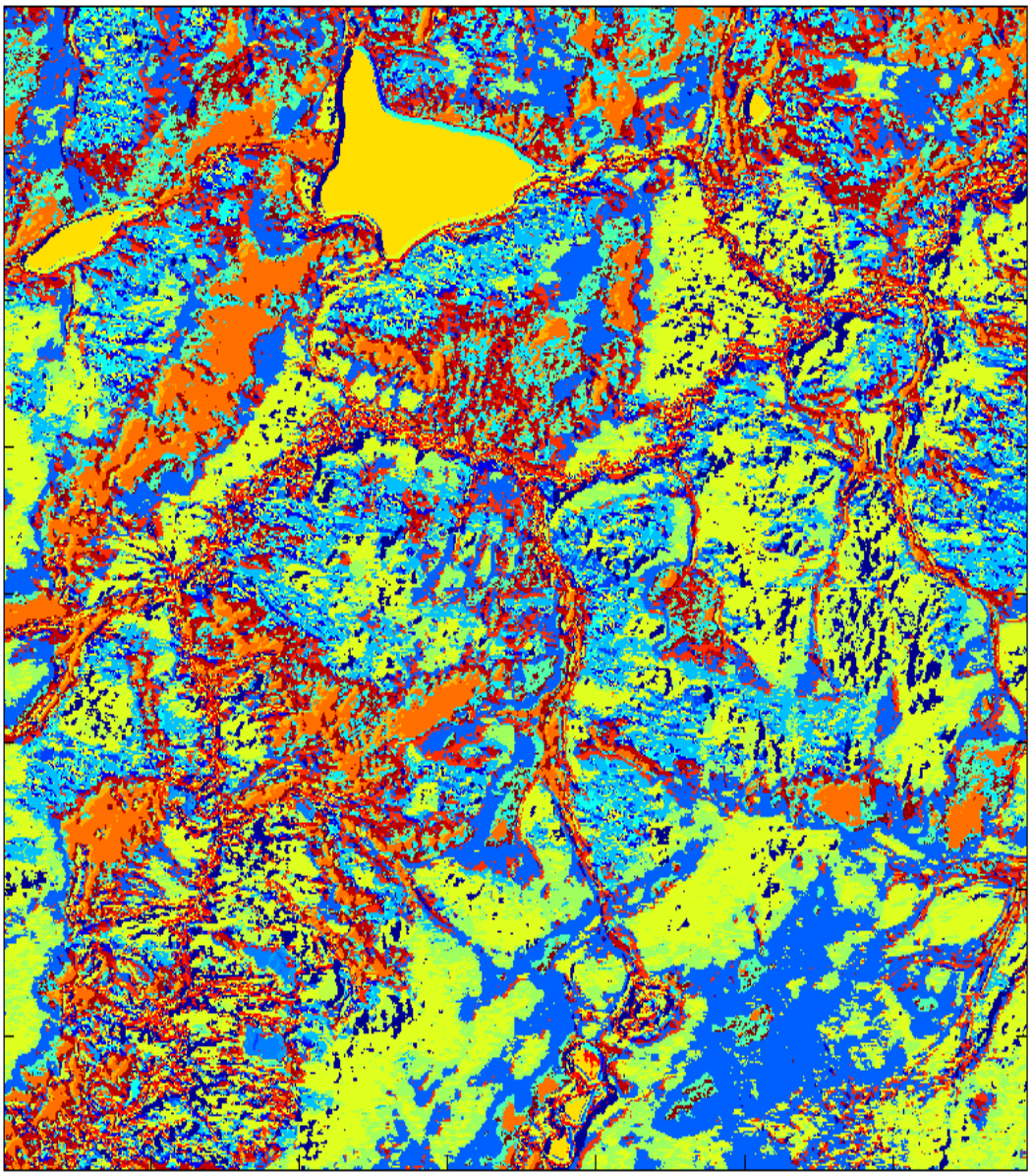


Figure 6.11: Category labels for 11x11 pixel patch. Analysis with 30 clusters with the larger spatial resolution leads to more coherent blocks of vegetation types. The non-vegetative textures are not detected as well in this case.

6.6 Conclusions on CoSA in satellite imagery

This chapter presented an example application of undercomplete learned dictionaries in an unsupervised fashion. These qualitative results use clustering of sparse approximations (CoSA) over learned dictionaries in satellite imagery processing, for the purpose of unsupervised land cover classification. The clustering process behaves as a classifier in detecting real variability, but some of the associations it makes could be lumped together or are not important for one type of classification versus another (e.g., vegetation analysis versus variability in the topography). Also, this chapter demonstrates that different spatial resolutions of the learned dictionary might be necessary depending on what specific features are desired in the analysis, i.e., aquatic, vegetative, or geomorphic. A multiscale approach could be successful in developing a classification scheme that shows vegetation at one scale, and topographic features at a different scale.

The challenge, therefore, to making the CoSA method successful for environmental studies will be to pre-condition the training set and the learning algorithm toward the features of interest, e.g., vegetation, by using indices such as NDVI and screening out spectral bands heavily influenced by moisture content, or by using LiDAR information to remove topographic features. The potential advantages are evident for the case where multiple classes are desired, e.g., transitional vegetation classes (areas that are not all shrub or all trees), where this method could be more efficient than trying to classify one class at a time using a subjective, supervised method.

7. Conclusion and Discussion

This thesis addresses several questions related to detection and classification of non-stationary RF signals in high noise environments using sparse techniques and dictionary methods, borrowing tools from several communities, including machine learning, applied mathematics, signal and image processing, computer vision, and computer science.

The research in this dissertation contributes to the extension of dictionary methods to RF classification in several ways. Most importantly, it demonstrates classification with ***undercomplete learned dictionaries***, which reduces dimensionality and helps make the goal of real-time processing feasible. Two algorithms for dictionary learning, Hebbian and K-SVD, are implemented and compared using a comprehensive parameter sensitivity analysis. A novel hybrid dictionary learning method is introduced and is shown to improve classification performance robustness to changes in SNR. The thesis also generalizes the use of undercomplete learned dictionaries by demonstrating unsupervised classification of land cover in multispectral satellite imagery. This chapter concludes the dissertation by summarizing the research of previous chapters and outlining future directions for this work in the near and long-term.

7.1 Precursor studies

An in-depth study of signal processing techniques used in transient and pulsed RF analysis was pursued prior to beginning work on an adaptive dictionary learning

software platform. This phase enabled a thorough analysis of the features obtained from an overcomplete parametric dictionary using an established algorithm and provided insight that motivated and shaped the learned dictionary work that followed. Specifically, the fast ridge pursuit algorithm published by Gribonval was implemented and optimized in Matlab. Despite its appeal of low computational complexity, the quality of features that could be directly extracted was found to be mediocre and this approach was eventually abandoned in favor of learned dictionary methods. It was, however, the seed thought that suggested perhaps a good dictionary for reconstruction is not necessarily a good dictionary for classification. This insight became central to the dissertation work and motivated the use of classification performance as quality metric in subsequent learned dictionary work. It was also the reason behind exploring undercomplete dictionaries for classification.

7.2 Study of learned dictionary methods

Dictionary learning has been a fast growing research field in the past two decades, and most of that growth has been in computer vision and image processing applications. Two learning algorithms were implemented, Aharon's K-SVD method [10] and a Hebbian learning method that was developed in this thesis and is similar to [15, 92, 96]. The classification task was cast as a two class problem (i.e., target signal ON or OFF) and the learned dictionaries were used in conjunction with a minimum residual classifier algorithm. Their classification performance on simulated RF data was compared and its dependence (and sensitivity) on the various learning parameters

(i.e., number of learning iterations, dictionary size, approximation sparsity) was evaluated. The study showed that for the representative RF signals chosen, optimally-learned Hebbian dictionaries can have higher discriminative power than K-SVD dictionaries and be more robust to small changes in SNR.

A practical challenge in RF signal processing is that the size of sampled data grows proportionally with the baseband frequency, shortening the processing time available for real applications. Dimensionality of data and of the analysis becomes a very important issue to consider and is the main reason why some signal processing approaches are impractical at high sample rates. From this perspective, the study on learned dictionary size yielded very encouraging results, in that it showed high performance with an undercomplete dictionary, and additionally with very coarse approximations (i.e., reduced dimensionality search).

For the K-SVD learning method, undercomplete dictionaries performed as well as complete or overcomplete dictionaries in classification. Also, it was shown that Hebbian undercomplete dictionaries outperform K-SVD learned dictionaries of any level of completeness. A novel method of learning hybrid Hebbian dictionaries was introduced and was shown to converge more rapidly to high classification accuracy, and improve classification performance stability across a wider range of learning iterations compared to the simple Hebbian dictionaries. A range of undercomplete Hebbian dictionaries, both simple and hybrid, was evaluated on two more complex simulated datasets, and the classification performance followed the same general patterns of parametric dependence previously observed. Noise

robustness was also evaluated by progressively reducing the amplitude of the target signals, and classification performance was compared to a STFT-based classifier. The learned dictionaries were found to give better robustness to SNR changes compared to a STFT-classifier, and this advantage may be worth the up-front learning computational expense.

An example of generalizing the use of undercomplete Hebbian dictionaries to different data was shown in Chapter 6 on unsupervised land cover classification in multispectral satellite imagery. The clustering of sparse approximations (CoSA) method showed very promising results and provided insights into developing a multiscale analysis tool for selective classification of landscape features.

The RF classification with learned dictionaries approach presented in this thesis demonstrated that high classification performance can not only be obtained with undercomplete dictionaries, but also by using coarser sparse approximations (smaller sparsity factor). The novelty of this signal processing approach is confirmed both by demonstrated functionality on realistic datasets, and also by the absence of similar concepts in the literature.

7.3 Future studies

The success of extending and adapting learned dictionary methods to RF signal processing motivates continued research in several important directions. This final section of the dissertation suggests additional studies that can make the use of undercomplete dictionaries more effective in other applications.

7.3.1 Modeling and estimating intrinsic data dimensionality

Most publications on dictionary learning assume it is necessary for the dictionary to be overcomplete with respect to the natural input signal dimensionality, i.e., the length of the training vectors. While that may be true, a more intuitive hypothesis presented in this thesis is that the learned dictionary size needs to be overcomplete with respect to some notion of ***intrinsic*** input dimensionality (i.e., the dimensionality of the underlying data space from which the inputs are sampled). Indeed, since a learned dictionary is seeking to approximate the underlying data manifold from which the training data are sampled, one would intuitively expect that the learned dictionary size depends on the amount of training data available and the intrinsic dimensionality of the data. Practically, it has been relatively common to simply choose a "sufficiently" overcomplete dictionary size with respect to the natural dimensionality (i.e., the length of input vectors). Empirically choosing the dictionary size has proven to be adequate for a variety of applications; however, a lower dimensionality bound has not been quantitatively explored. This intrinsic dimensionality is practically hard to evaluate, and setting a dictionary size few times larger than the natural dimensionality has yielded good results in recent work. A model is currently being developed to help estimate the intrinsic dimensionality of training data and inform the selection of learned dictionary size.

7.3.2 Exploring other sparsifying norms

The sparse approximations that were used throughout this thesis were found using an l_0 "norm." This allows an easier, faster implementation, as well as a progressive way of increasing the sparsity of a particular approximation to study effects on classification performance. One of the findings in this thesis was that sparser approximations (i.e., using a smaller number of coefficients) had a positive impact on classification. A potential extension of the work would incorporate the use of regularized fractional l_p norm [147] that has been shown to provide even sparser approximations with smaller reconstruction error compared to the other sparsifying norms, in particular the much used l_1 norm. While this method admittedly introduces additional complexity, it could allow faster dictionary learning by using better sparse approximations of the training data, and may lead to better classification performance using undercomplete dictionaries.

7.3.3 Change detection in satellite imagery using CoSA

The application of undercomplete Hebbian dictionaries to multispectral satellite image analysis has opened a number of possible immediate avenues for research in the area of change detection in images (in this case, of the Arctic environment). Both seasonal (e.g., vegetation growth in the summer) and longer term (e.g., extent of shrub cover in July 2009 vs. July 2012) changes could be detected using the ***CoSA*** (clustering of sparse approximations) method and would provide environmental experts with quantitative data for climate change models. Also, two

other locations in the Arctic are currently exhibiting significant hydrologic changes, and automatic high resolution classification of landscape features is becoming a necessity. A way to make the CoSA method more successful for environmental studies is to pre-condition the learning toward the features of interest, e.g., vegetation, by using indices such as NDVI to screen out bands heavily influenced by moisture content. Additionally, data fusion dictionaries could be learned using multispectral data enhanced with LiDAR information to provide topographic features.

7.3.4 Lightning research

An extension of thesis work to a real problem is applying machine learning techniques to lightning classification. This research focuses on modeling and analysis of signals for orbital remote classification of lightning data under LANL's space research program. For over two decades the program has included an active research effort utilizing satellite observations of terrestrial lightning to learn more about the Earth's RF background. Arguably the richest satellite lightning database ever recorded is that from the Fast On-orbit Recording of Transient Events (FORTE) satellite, which was launched in 1997 and returned at least five years of data from its two payloads. The LANL FORTE RF database remains relevant for the application of modern event classification techniques to further lightning research in the scientific community. Some classification work has been done previously using on-orbit databases, but the focus has been primarily on developing physical models and understanding the effect of the ionosphere on orbital recordings. The field is ripe for

scientific discovery, and application of the methods developed in this thesis would significantly impact the community.

7.4 Closing remarks

Dictionary methods are regarded as an enabling approach to processing signals that are not easily characterized by analytical models. They provide adaptive representations that can be very sparse, but due to their lack of orthogonality and need of specific search algorithms, can incur high computational costs. This dissertation focused primarily on extending and adapting learned dictionary techniques to RF classification, and demonstrated practical, high performance dictionary implementations that are undercomplete. It has the potential to introduce the RF community to machine learning techniques and enable wider use in a host of practical applications.

Bibliography

[1] A. V. Oppenheim, "Digital-Time Signal Processing," in *Signal Processing*, 2nd ed: Prentice Hall, 1999.

[2] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, pp. 713-718, 1992.

[3] R. E. Learned, *et al.*, "Wavelet packet based transient signal classification," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, 1992, pp. 109-112.

[4] M. P. Caffrey and S. D. Briles, "Space-based RF signal classification using adaptive wavelet features," in *Proceedings of SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition IV*, 1995, pp. 433-442.

[5] S. G. Mallat, *A Wavelet Tour of Signal Processing. The sparse way*, 3rd ed.: Academic Press, 2009.

[6] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397-3415, 1993.

[7] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Transactions on Signal Processing*, vol. 49, pp. 994-1001, 2001.

[8] S. P. Brumby, *et al.*, "Capturing dynamics on multiple time scales: a multilevel fusion approach for cluttered electromagnetic data," in *Proc SPIE Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, 2010.

[9] B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," *Network-Computation in Neural Systems*, vol. 7, pp. 333-339, 1996.

[10] M. Aharon, *et al.*, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4311-4322, 2006.

[11] K. Engan, *et al.*, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings Vols I-VI*, 1999, pp. 2443-2446.

[12] B. Mailhé, *et al.*, "Shift-invariant dictionary learning for sparse representations: extending K-SVD," in *EUSIPCO*, 2008.

[13] S. P. Brumby, "Image fusion for remote sensing using fast, large-scale neuroscience models," in *Proceedings of SPIE*, Orlando, FL, 2011.

[14] J. Mairal, *et al.*, "Discriminative learned dictionaries for local image analysis," in *IEEE Conference on Computer Vision and Pattern Recognition, Vols 1-12*, 2008, pp. 2415-2422.

[15] K. Skretting and J. H. Husøy, "Texture classification using sparse frame-based representations," *Eurasip Journal on Applied Signal Processing*, 2006.

[16] D. I. Moody, *et al.*, "Classification of transient signals using sparse representations over adaptive dictionaries," in *Proceedings of SPIE Conference on Independent Component Analyses, Wavelets, Neural Networks, Biosystems, and Nanoengineering IX*, 2011.
[17] D. I. Moody, *et al.*, "Sparse classification of RF transients using chirplets and learned dictionaries," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2011, pp. 1888-1892.
[18] D. I. Moody, *et al.*, "Radio frequency (RF) transient classification using sparse representations over learned dictionaries," in *Proceedings of SPIE Conference on Wavelets and Sparsity XIV*, 2011.
[19] D. I. Moody, *et al.*, "Learning sparse discriminative representations for land cover classification in the Arctic," in *(to appear in) SPIE Satellite Data Compression, Communications, and Processing VIII*, San Diego, CA, 2012.
[20] D. I. Moody, *et al.*, "Arctic land cover classification using multispectral imagery with adaptive sparse representations," presented at the Conference on Data Analysis, Santa Fe, 2012.
[21] R. O. Duda, P. Hart, D. Stork, *Pattern Classification*, 2nd ed.: John Wiley & Sons, Inc., 2001.
[22] G. L. Turin, "An introduction to matched-filters," *IRE Transactions on Information Theory*, vol. 6, pp. 311-329, 1960.
[23] G. L. Turin, "Introduction to digital matched-filters," *Proceedings of the IEEE*, vol. 64, pp. 1092-1112, 1976.
[24] D. Gabor, "Theory of communication," *J. Inst. Elec. Eng.*, vol. 93, pp. 429-457, 1946.
[25] S. G. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed.: Academic Press, 1999.
[26] A. Graps, "An introduction to wavelets," *IEEE Computational Sciences and Engineering*, vol. 2, pp. 50-61, 1995.
[27] A. Grossman and J. Morlet, "Decompositions of Hardy functions into square integrable wavelets of constant shape," *SIAM J. Math*, vol. 15, pp. 723-736, 1984.
[28] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, 1989.
[29] I. Daubechies, "Ten lectures on wavelets," in *Number 61 in CBMS-NSF Series in Applied Mathematics*. Philadelphia: SIAM Publications, 1992.
[30] R. Coifman, *et al.*, "Adapted wave form analysis, wavelet-packets and applications," in *ICIAM 91*, Washington, DC, 1991, pp. 41-50.
[31] M. V. Wickerhauser, *Adapted Wavelet Analysis: From Theory to Software*: A K Peters/CRC Press, 1996.
[32] G. Strang and T. Nguyen, "Wavelets and Filter Banks," 2nd ed: Wellesley College, 1996.

[33] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory,* vol. 41, pp. 613-627, 1995.
[34] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice (The International Series in Engineering and Computer Science).* Norwell, MA: Kluwer: Springer, 2002.
[35] J. Bradley, *et al.*, "The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," in *Tech. Report LA-UR-93-1659*. Los Alamos, NM, 1993.
[36] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.,* vol. 41, pp. 906-966, 1988.
[37] M. Frisch and H. Messer, "Transient signal detection using prior information in the likelihood ratio test," *IEEE Transactions on Signal Processing,* vol. 41, 1993.
[38] M. Desai and D. J. Shazeer, "Acoustic transient analysis using wavelet decomposition," in *IEEE Conference on Neural Networks for Ocean Engineering*, 1991, pp. 29-40.
[39] R. E. Learned and A. S. Willsky, "A wavelet packet approach to transient signal classification," *Applied and Computational Harmonic Analysis,* vol. 2, pp. 265-278, 1995.
[40] W. Sweldens, "The lifting scheme: a construction of second generation wavelets," *SIAM J. MATH. ANAL.*, vol. 29, pp. 511-546, 1998.
[41] C. M. Brislawn and B. Wohlberg, "Gain normalization of lifted filter banks," *Signal Processing,* vol. 87, pp. 1281-1287, 2007.
[42] I. W. Selesnick, *et al.*, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine,* vol. 22, pp. 123-151, 2005.
[43] R. Baraniuk, "Optimal tree approximation with wavelets," in *Wavelet Applications in Signal and Image Processing VII*, 1999, pp. 196-207.
[44] T. Hastie, *et al.*, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.: Springer, 2009.
[45] S. M. Stigler, *The history of statistics, Chapter 3*: Harvard University press, 1986.
[46] R. L. Wolpert, "A conversation with James O. Berger," *Statistical Science,* vol. 19, pp. 205-218, 2004.
[47] L. Angrisani, *et al.*, "Wavelet-network-based detection and classification of transients," *IEEE Transactions on Instrumentation and Measurement,* vol. 50, pp. 1425-1435, 2001.
[48] M. S. Crouse, *et al.*, "Wavelet-based statistical signal processing," *IEEE Transactions on Signal Processing,* vol. 46, pp. 886-902, 1998.
[49] C. H. Lee, *et al.*, "A literature survey of wavelets in power engineering applications," *Proc. Natl. Sci. Counc. ROC(A),* vol. 24, pp. 249-253, 2000.
[50] N. Perera, *et al.*, "Development of an on-line transient classification system," in *International Conference on Power Systems Transients*, Kyoto, Japan, 2009.

[51] D. G. Stork and E. Yom-Tov, *Computer manual in MATLAB to accompany Pattern Classification, 2nd Ed.*: Wiley, 2004.
[52] C. Chang and C. Lin, "LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, pp. 1-27, 2011.
[53] M. Hall, *et al.*, "The WEKA data mining software: an update," *SIGKDD Explorations,* vol. 11, 2009.
[54] Y. Shin, *et al.*, "Design of a time-frequency domain matched filter for detection of non-stationary signals," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vols I-Vi, Proceedings*, 2001, pp. 3585-3588.
[55] S. Briles, *et al.*, "Innovative use of DSP technology in space: FORTE event classifier," in *Proceedings of the International Workshop on Artifcial Intelligence in Solar-Terrestrial Physics*, 1993.
[56] D. Eads, *et al.*, "Genetic algorithms and support vector machines for time series classification," *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V,* vol. 4787, pp. 74-85, 2002.
[57] J. J. Benedetto and S. D. Li, "The theory of multiresolution analysis frames and applications to filter banks," *Applied and Computational Harmonic Analysis,* vol. 5, pp. 389-427, 1998.
[58] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed.: Springer, 2010.
[59] I. Daubechies, "Time frequency localization operators - a geometric phase-space approach," *IEEE Transactions on Information Theory,* vol. 34, pp. 605-612, 1988.
[60] Y. C. Pati, *et al.*, "Orthogonal matching pursuit - recursive function approximation with applications to wavelet decomposition," in *IEEE Asilomar Conference on Signals, Systems & Computers, Vols 1 and 2*, 1993, pp. 40-44.
[61] S. S. B. Chen, *et al.*, "Atomic decomposition by basis pursuit," Stanford University, 1995.
[62] S. S. B. Chen, *et al.*, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, pp. 129-159, 2001.
[63] S. Levy and P. K. Fullagar, "Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution," *Geophysics,* vol. 46, pp. 1235-1243, 1981.
[64] E. J. Candes, *et al.*, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, pp. 489-509, 2006.
[65] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Transactions on Information Theory,* vol. 52, pp. 5406-5425, 2006.
[66] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory,* vol. 52, pp. 1289-1306, 2006.

[67] R. Chartrand, "Nonconvex regularization for shape preservation," in *2007 IEEE International Conference on Image Processing, Vols 1-7*, 2007, pp. 293-296.
[68] R. Chartrand, "Nonconvex compressed sensing and error correction," in *2007 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol III*, 2007, pp. 889-892.
[69] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Vols 1 and 2*, 2009, pp. 262-265.
[70] M. Elad, *et al.*, "On the role of sparse and redundant representations in image processing," *IEEE Proceedings - Special Issue on Applications of Sparse Representation & Compressive Sensing*, vol. 98, pp. 972-982, 2010.
[71] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 58, pp. 267-288, 1996.
[72] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, pp. 3736-3745, 2006.
[73] R. Rubinstein, *et al.*, "Double sparsity: learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1553-1564, 2010.
[74] J. Mairal, *et al.*, "Non-local sparse models for image restoration," in *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, 2009, pp. 2272-2279.
[75] J. Mairal, *et al.*, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, pp. 53-69, 2008.
[76] J. Mairal, *et al.*, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling & Simulation*, vol. 7, pp. 214-241, 2008.
[77] J. Zepeda, *et al.*, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 1061-1073, 2011.
[78] I. Tosic, *et al.*, "Ultrasound tomography with learned dictionaries," in *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 5502-5505.
[79] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled K-space data by dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 30, pp. 1028-1041, 2011.
[80] R. Rigamonti, *et al.*, "Are sparse representations really relevant for image classification?," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1545-1552.

[81] J. Mairal, *et al.*, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Computer Vision - ECCV 2008, Pt III, Proceedings*, 2008, pp. 43-56.
[82] A. S. Charles, *et al.*, "Learning sparse codes for hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing,* vol. 5, pp. 963-978, 2011.
[83] I. Daubechies, "From the original framer to present-day time-frequency and time-scale frames," *Journal of Fourier Analysis and Applications,* vol. 3, pp. 485-486, 1997.
[84] J. L. Starck, *et al.*, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing,* vol. 11, pp. 670-684, 2002.
[85] E. J. Candes and D. L. Donoho, "Curvelets and reconstruction of images from noisy Radon data," in *Wavelet Applications in Signal and Image Processing VIII*, 2000, pp. 108-117.
[86] K. H. Guo and D. Labate, "Sparse shearlet representation of Fourier integral operators," *Electronic Research Announcements in Mathematical Sciences,* vol. 14, pp. 7-19, 2007.
[87] D. Labate and K. H. Guo, "Optimally sparse shearlet approximations of 3D data," in *Independent Component Analyses, Wavelets, Neural Networks, Biosystems, and Nanoengineering IX*, 2011.
[88] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, pp. 2091-2106, 2005.
[89] E. J. Candes and D. L. Donoho, "Ridgelets: a key to higher-dimensional intermittency?," *Philosophical Transactions of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences,* vol. 357, pp. 2495-2509, 1999.
[90] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing,* vol. 14, pp. 423-438, 2005.
[91] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607-609, 1996.
[92] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Research,* vol. 37, pp. 3311-3325, 1997.
[93] H. Lee, *et al.*, "Unsupervised learning of hierarchical representations with convolutional deep belief networks," *Communications of the ACM,* vol. 54, pp. 95-103, 2011.
[94] B. A. Olshausen, "Learning sparse, overcomplete representations of time-varying natural images," in *2003 International Conference on Image Processing, Vol 1, Proceedings*, 2003, pp. 41-44.

[95] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011, pp. 1517-1520.
[96] S. P. Brumby, *et al.*, "Large-scale functional models of visual cortex for remote sensing," in *38th IEEE Applied Imagery Pattern Recognition, Vision: Humans, Animals, and Machines*, Cosmos Club, Washington DC 2009.
[97] J. Wright, *et al.*, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210-227, 2009.
[98] J. Wright, *et al.*, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, pp. 1031-1044, 2010.
[99] J. Mairal, *et al.*, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19-60, 2010.
[100] R. Jenatton, *et al.*, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297-2334, 2011.
[101] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
[102] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145-175, 2001.
[103] S. Lazebnik, *et al.*, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
[104] J. C. Yang, *et al.*, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition, Vols 1-4*, 2009, pp. 1794-1801.
[105] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 2, Proceedings*, 2005, pp. 524-531.
[106] H. Zhang, *et al.*, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *CVPR*, 2006.
[107] J. Deng, *et al.*, "What does classifying more than 10,000 image categories tell us?," in *Computer Vision-ECCV 2010, Pt V*, 2010, pp. 71-84.
[108] J. Mairal, "Sparse coding for machine learning, image processing and computer vision," PhD, Mathématiques Appliquées, L'École Normale Supérieure de Chacan, Paris, 2010.
[109] J. Mairal, *et al.*, "Learning hierarchical and topographic dictionaries with structured sparsity," in *Wavelets and Sparsity XIV*, 2011.
[110] J. Mairal, *et al.*, "Multiscale sparse image representation with learned dictionaries," in *2007 IEEE International Conference on Image Processing, Vols 1-7*, 2007, pp. 1233-1236.

[111] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *International Conference on Machine Learning*, 2011, pp. 921-928.
[112] B. L. Sturm and M. G. Christensen, "Cyclic matching pursuits with multiscale time-frequency dictionaries," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2011.
[113] E. J. Candes, *et al.*, "Enhancing sparsity by reweighted l(1) minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877-905, 2008.
[114] S. Krstulovic and R. Gribonval, "MPTK: Matching pursuit made tractable," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, Vols 1-13*, 2006, pp. 2947-2950.
[115] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2370-2382, 2008.
[116] B. Mailhe, *et al.*, "A low complexity orthogonal matching pursuit for sparse signal approximation with shift-invariant dictionaries," in *2009 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vols 1- 8, Proceedings*, 2009, pp. 3445-3448.
[117] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9, pp. 137-140, 2002.
[118] M. G. Christensen and S. H. Jensen, "The cyclic matching pursuit and its application to audio modeling and coding," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2007.
[119] R. G. Baraniuk and D. L. Jones, "Shear madness - new orthonormal bases and frames using chirp functions," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3543-3549, 1993.
[120] R. Gribonval, "Approximations non-linéaires pour l'analyse des signaux sonores," PhD, Mathématiques Appliquées, L'Université de Paris IX Dauphne, 1999.
[121] D. O. Hebb, *et al.*, "The organization of behavior - a neuropsychological theory," *Contemporary Psychology*, vol. 39, pp. 1018-1020, 1994.
[122] R. Quinlan, *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.
[123] N. R. Harvey, *et al.*, "Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 393-404, 2002.
[124] S. P. Brumby, *et al.*, "Evolving feature extraction algorithms for hyperspectral and fused imagery," in *Proceedings of the Fifth International Conference on Information Fusion, Vol II*, 2002, pp. 986-993.
[125] N. R. Harvey, *et al.*, "Automated simultaneous multiple feature classification of MTI data," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII*, 2002, pp. 346-356.
[126] N. R. Harvey, *et al.*, "Parallel evolution of image processing tools for multispectral imagery," in *Imaging Spectrometry VI*, 2000, pp. 72-82.

[127] J. Theiler and D. M. Cai, "Resampling approach for anomaly detection in multispectral images," in *Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery IX*, 2003, pp. 230-240.
[128] J. Theiler and G. Gisler, "A contiguity-enhanced *k*-means clustering algorithm for unsupervised multispectral image segmentation," in *Algorithms, Devices, and Systems for Optical Information Processing*, 1997, pp. 108-118.
[129] "ACIA, 2005.Arctic climate impact assessment. Cambridge University Press, 1046p."
[130] M. Sturm, *et al.*, "Snow-shrub interactions in Arctic tundra: A hypothesis with climatic implications," *Journal of Climate*, vol. 14, pp. 336-344, 2001.
[131] C. R. Burn and S. V. Kokelj, "The environment and permafrost of the Mackenzie delta area," *Permafrost and Periglacial Processes*, vol. 20, pp. 83-105, 2009.
[132] T. C. Lantz and S. V. Kokelj, "Increasing rates of retrogressive thaw slump activity in the Mackenzie Delta region, NWT, Canada," *Geophysical Research Letters*, vol. 35, 2008.
[133] P. Marsh, *et al.*, "Snowmelt energetics at a shrub tundra site in the western Canadian Arctic," *Hydrological Processes*, vol. 24, pp. 3603-3620, 2010.
[134] H. E. Epstein, *et al.*, "Detecting changes in arctic tundra plant communities in response to warming over decadal time scales," *Global Change Biology*, vol. 10, pp. 1325-1334, 2004.
[135] D. A. Stow, *et al.*, "Remote sensing of vegetation and land-cover change in Arctic tundra ecosystems," *Remote Sensing of Environment*, vol. 89, pp. 281-308, 2004.
[136] T. C. Lantz, *et al.*, "Response of green alder (Alnus viridis subsp fruticosa) patch dynamics and plant community composition to fire and regional temperature in north-western Canada," *Journal of Biogeography*, vol. 37, pp. 1597-1610, 2010.
[137] T. C. Lantz, *et al.*, "Spatial heterogeneity in the shrub tundra ecotone in the Mackenzie delta region, Northwest territories: implications for Arctic environmental change," *Ecosystems*, vol. 13, pp. 194-204, 2010.
[138] I. Olthof, *et al.*, "Recent (1986-2006) vegetation-specific NDVI trends in northern Canada from satellite data," *Arctic*, vol. 61, pp. 381-394, 2008.
[139] M. D. Walker, *et al.*, "Plant community responses to experimental warming across the tundra biome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, pp. 1342-1346, 2006.
[140] T. N. Nguyen, *et al.*, "Estimating the extent of near-surface permafrost using remote sensing, Mackenzie delta, Northwest territories," *Permafrost and Periglacial Processes*, vol. 20, pp. 141-153, 2009.
[141] A. G. Lewkowicz and C. Harris, "Frequency and magnitude of active-layer detachment failures in discontinuous and continuous permafrost, northern Canada," *Permafrost and Periglacial Processes*, vol. 16, pp. 115-130, 2005.

[142] K. M. Hinkel, *et al.*, "Spatial extent, age, and carbon stocks in drained thaw lake basins on the Barrow Peninsula, Alaska," *Arctic Antarctic and Alpine Research,* vol. 35, pp. 291-300, 2003.
[143] G. Hugelius, *et al.*, "High-resolution mapping of ecosystem carbon storage and potential effects of permafrost thaw in periglacial terrain, European Russian Arctic," *Journal of Geophysical Research-Biogeosciences,* vol. 116, 2011.
[144] R. B. Myneni, *et al.*, "The interpretation of spectral vegetation indexes," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 33, pp. 481-486, 1995.
[145] N. R. Harvey, *et al.*, "Finding golf courses: The ultra high tech approach," *Real-World Applications of Evolutionary Computing, Proceedings,* vol. 1803, pp. 54-64, 2000.
[146] DigitalGlobe®. (2010). *The benefits of the 8 spectral bands of WorldView-. Available online: http://worldview2.digitalglobe.com/about/.*
[147] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters,* vol. 14, pp. 707-710, 2007.