An Analysis of Sparseness and Regularization in Exemplar-Based Methods for Speech Classification

Dimitri Kanevsky, Tara N. Sainath, Bhuvana Ramabhadran, David Nahamoo

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A
{kanevsky, tsainath, bhuvana, nahamoo}@ibm.com

Abstract

The use of exemplar-based techniques for both speech classification and recognition tasks has become increasingly popular in recent years. However, the notion of why sparseness is important for exemplar-based speech processing has been relatively unexplored. In addition, little analysis has been done in speech processing on the appropriateness of different types of sparsity regularization constraints. The goal of this paper is to answer the above two questions, both through mathematically analyzing different sparseness methods and also comparing these approaches for phonetic classification in TIMIT.

1. Introduction

Sparse Representation techniques for machine learning applications have become increasing popular in recent years (i.e., [2], [4], [7]). Exemplar-based methods, including k-nearest neighbors (kNN) [2], support vector machines (SVMs) and sparse representations (SRs) [7], utilize the details of actual training examples when making a classification decision. Since the number of training examples in speech tasks can be very large, many of these exemplar-based techniques adopt a common theme of using a few number of training examples to characterize a test vector. This is in contrast to standard regression methods, such as ridge regression [10], nearest subspace [11] and nearest line [11] techniques, which utilize information about all training examples when characterizing a test vector.

Since speech itself is not a sparse signal, the notion of why sparse representations are important has garnered little attention thus far in the speech community. In addition, a variety of different sparseness techniques, which employ different types of regularization, have been explored for speech tasks [4], [7]. However, little work has been done in comparing the difference in these methods for speech processing tasks.

Thus, the first goal of this paper is to analyze the difference between exemplar-based classification with and without enforcing sparseness. First, we mathematically analyze the differences between the sparse representation and ridge regression techniques. Next, we analyze the behavior of the two methods on a practical speech task. Namely, we explore difference in classification performance between the two techniques for phonetic classification in TIMIT [6]. While this paper only explore sparseness techniques for classification, we argue that good classification is fundamental for good recognition performance. For example, as [8] shows, having a SR method with high frame classification accuracy can often translate into improvements in recognition performance as well.

The second goal of this paper is to explore if sparseness is used in speech processing, what type of sparseness regularization should be employed? Typically sparseness methods such as LASSO [9] and Bayesian Compressive Sensing (BCS) [5] use an $l_1$ sparseness constraint (known as a Laplacian prior). This is in contrast to an Elastic Net [12], which uses a combination of an $l_1$ and $l_2$ (Gaussian prior) constraint. Similarly, Approximate Bayesian Compressive Sensing (ABCS) [7] uses an $l_1^2$ constraint, which is known as a Semi-Gaussian prior. First, we mathematically analyze the difference in the sparseness objectives for the above methods. Next, we compare the performance of these methods for phonetic classification in TIMIT.

This paper is organized as follows. Section 2 describes the main framework behind exemplar-based classification. Section 3 gives a brief description of the TIMIT corpus. Section 4 analyzes why sparseness can be useful in classification tasks. Section 5 compares the performance of different sparseness methods for classification. Finally, Section 6 concludes the paper and discusses future work.

2. Exemplar-Based Methods for Classification

In this section, we describe a framework to classify a test point using examples from the training set, and then highlight two exemplar-based methods which fall into this framework.

2.1. Classification Based on Exemplars

The goal of classification is to use training data from $k$ different classes to determine the best class to assign to test vector $y$. First, let us consider taking all training examples $n_i$ from class $i$ and concatenate them into a matrix $H_i$ as columns, in other words $H_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, where $x \in \mathbb{R}^m$ represents a feature vector from the training set of class $i$ with dimension $m$. Given sufficient training examples from class $i$, [11] shows that a test sample $y$ from the same class can be represented as a linear combination of the entries in $H_i$ weighted by $\beta$, that is:

$$y = \beta_{i,1} x_{i,1} + \beta_{i,2} x_{i,2} + \ldots + \beta_{i,n_i} x_{i,n_i}. \quad (1)$$

However, since the class membership of $y$ is unknown, we define a matrix $H$ to include training examples from all $k$ classes in the training set, in other words the columns of $H$ are defined as $H = [H_1, H_2, \ldots, H_k] = [x_{1,1}, x_{1,2}, \ldots, x_{k,n_k}] \in \mathbb{R}^{m \times N}$. Here $N$ is the total number of all training examples from all classes. We can then write test vector $y$ as a linear combination of all training examples, in other words $y = H \beta$. We can solve this linear system for $\beta$ and use information about $\beta$ to make a classification decision. Specifically, large entries of $\beta$ should correspond to the entries in $H$ with the same class as $y$. Thus, one proposed classification decision approach [7] is to compute the $l_2$ norm for all $\beta$ entries within a specific class, and choose the class with the largest $l_2$ norm support.
2.2. Exemplar-Based Methods

Various types of exemplar-based classifiers can be cast in the framework of representing the test vector \( y \) as a linear combination of training examples \( H \), subject to a constraint on \( \beta \). Below, we review two popular techniques that are based on the following optimization problem for various values of \( q \) and \( \alpha \)

\[
\min_{\beta} \| y - H \beta \|_2 \text{ s.t. } \| \beta \|_q^\alpha < \epsilon
\]

2.2.1. Ridge Regression

Ridge regression (RR) methods [10] use information about all training examples in \( H \) to make a classification decision about \( y \), in contrast to a nearest-neighbor (NN) approach to exemplar-based classification, which uses information about just 1 training example. Specifically, the RR method looks to project \( y \) into the linear space of all training examples and solves for the \( \beta \) which minimizes Equation 2 for \( q = 2, \alpha = 2 \). The term \( \| \beta \|_2 < \epsilon \) is an \( l_2 \) norm on \( \beta \) (i.e., a Gaussian constraint) but does not enforce any sparseness.

2.2.2. Sparse Representations

Like RR methods, sparse representation (SR) techniques (i.e., [7], [11]), project \( y \) into the linear span of examples in \( H \), but constrain \( \beta \) to be sparse. Specifically, SR methods solve for \( \beta \) by minimizing Equation 2, given various settings for \( \alpha \) and \( q \). For example, in a probabilistic setting \( q = 1, \alpha = 1 \) leads to a Laplacian constraint, whereas \( q = 1, \alpha = 2 \) leads to a Semi-Gaussian constraint. The remainder of this paper is focused on comparing the RR method to various SR methods with different types of regularizations.

3. Experiments

We analyze the behavior of various exemplar-based methods on the TIMIT [6] corpus. The corpus contains over 6,300 phonetically rich utterances divided into three sets, namely the training, development, and core test set. For testing purposes, the standard practice is to collapse the 48 trained labels into a smaller set of 39 labels. All methods are tuned on the development set and all experiments are reported on the core test set.

The complete experimental setup, as well as the features used for classification, are similar to [7]. First, we represent each frame in our signal by a 40 dimensional discriminatively trained Space Boosted Maximum Mutual Information (DBMMI) feature. We split each phonetic segment into thirds, taking the average of these frame-level features around 3rds, and splice them together to form a 120 dimensional vector. This allows us to capture time dynamics into each segment. Then, at each segment, segmental feature vectors to the left and right of this segment are joined together and a Linear Discriminative Analysis (LDA) transform is applied to project 200 dimensional feature vector down to 40 dimensions.

Similar to [7], we find a neighborhood of closest points to \( y \) in the training set using a kd-tree. These \( k \) neighbors become the entries of \( H \). We explore classification performance for different sizes of \( H \).

In this paper, we explore the following two questions, using TIMIT to provide experimental results to support our framework.

- Why and when is sparseness important for exemplar-based methods?
- If sparseness is used, what type of regularization constraint should be utilized?

4. Why Sparse Representations?

4.1. Motivation

The SR method can be considered a generalization of the NN and RR methods. NN classifies a test sample \( y \) based on the class of a single training sample, whereas RR finds the best linear representation for \( y \) in terms of all the training samples in \( H \) since little sparseness is enforced on \( \beta \). The SR classifier is more generalizable than the NN and RR methods. Similar to RR, it uses a multiple (instead of one) training samples in \( H \) to linearly represent the test sample. However, it uses a small number of training samples in \( H \) compared to RR. Thus, as [11] shows, the SR classifier can better adapt to the actual distributions of the training samples and is less sensitive to outliers and over-fitting.

We will motivate the difference between the RR and SR methods further with the following example. Let us consider a \( 2 \times 7 \) matrix \( H = \begin{bmatrix} h_1, h_2, h_3, h_4, h_5, h_6, h_7 \end{bmatrix} \)

\[
\begin{bmatrix}
0.2 & 0.1 & 0.4 & 0.3 & -0.6 & 0.6 & -0.6 \\
0.2 & 0.3 & 0.35 & 0.3 & 0.1 & 0.3 & 0.4
\end{bmatrix}
\]

where first three columns \( h_1, h_2, h_3 \) are “training” utterances that belong to a class \( C_1 \) and last four columns are “training” utterances that belong \( C_2 \). Assume also that a vector \( y = [0.29; 0.29] \) is “test” data that belong to a class \( C_1 \).

4.2. Results

To analyze the behavior of the SR and RR methods in a practical speech example, we explore phonetic classification on TIMIT as the size of \( H \) is varied from 1 to 10,000. A plot of the error rate for the two methods for varied \( H \) is shown Figure 2. For
this figure, we again used the ABCS SR method. First notice that as the size of $H$ increases up to 1,000 the error rates of the RR and SR both decrease, showing the benefit of including multiple training examples when making a classification decision. Also notice that there is no different in error between the RR and SR both decrease, showing the benefit of including sparseness to select only a few examples in $H$ to explain $y$ rather than all examples in $H$.

5. What Type of Regularization?

Now that we have motivated the use of regularization, in this section we analyze different forms of regularization. As illustrated in Equation 2, with $q = 1$, a sparse representation solution can be formulated by finding the $\beta$ which minimizes the residual error $\| y - H \beta \|_2$, subject to a regularization $\| \beta \|_q < \epsilon$ on $\beta$. There are four common types of regularizations on $\beta$.

- If $q = 2$ and $\alpha = 2$, then the regularization becomes $\| \beta \|_2 < \epsilon$. This constraint can be modeled as a Gaussian prior. Common techniques which impose an $l_2$ constraint on $\beta$ include Ridge Regression [10].

- If $q = 1$ and $\alpha = 1$, then the regularization becomes $\| \beta \|_1 < \epsilon$. This constraint can be modeled a Laplacian prior. Common techniques which impose an $l_1$ constraint on $\beta$ include LASSO [9] and Bayesian Compressive Sensing (BCS) [5]. These techniques will be discussed in more detail below.

- Many techniques also impose a combination of an $l_1$ and $l_2$ constraint on $\beta$. These methods include Elastic Net [12] and Cyclic Subgradient Projections (CSP) 5.

- The Approximate Bayesian Compressive Sensing (ABCS) [7] approach imposes a semi-Gaussian constraint, which can be represented as $\| \beta \|_1^2 < \epsilon$. The motivation for this semi-Gaussian constraint will be discussed in more detail below.

5.1. Regularization Methods

5.1.1. LASSO

The LASSO problem imposes an $l_1$ constraint on $\beta$. The LASSO problem is equivalent to Equation 2 for $q = 1$, $\alpha = 1$. It can be formulated as follows:

$$
\min_{\beta} \| y - H \beta \|_2 + \lambda \| \beta \|_1
$$

Here $\lambda$ controls the weight of the $l_1$ norm. The Least Angle Regression (LARS) ([3]) solves LASSO through a forward stepwise regression, computing point estimates of $\beta$ at each step.

The effect of the $l_2$ norm is to spread values of entries in $\beta$ equally. Therefore the optimization problem in Equation 2 for $q = 2$ tries to find a balance between keeping the residual $\| y - \beta \|_2$ small and trying to keep all the entries in the vector $\beta$ to be non-zero. In contrast, the norm $l_1$ tries to enforce sparsity $\beta$ while keeping the residual $\| y - H \beta \|_2$ small.

5.1.2. Bayesian Compressive Sensing

Bayesian Compressive sensing [5] also uses an $l_1$ constraint on $\beta$. It can be formulated similarly to Equation 3. BCS introduces a probabilistic framework to estimate the sparseness parameters required for signal recovery. This technique limits the effort required to tune the sparseness constraint, and also provides complete statistics for the estimate of $\beta$.

5.1.3. Elastic Net

The Elastic Net [12] method imposes a mixture of an $l_1$ and $l_2$ constraint on $\beta$. It can be formulated as follows:

$$
\min_{\beta} \| y - H \beta \|_2 + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|_2^2
$$

Here $\lambda_1$ and $\lambda_2$ are the weights controlling the $l_1$ and $l_2$ constraint. In the elastic net formulation, the $l_1$ term enforces the sparsity of the solution, whereas the $l_2$ penalty ensures democracy among groups of correlated variables. The second term has also a smoothing effect that stabilizes the obtained solution.

5.1.4. CSP

The Cyclic Subgradient Projection technique [11] also imposes a mixture of an $l_1$ and $l_2$ constraint on $\beta$. The solution of $\beta$ is found using a sequence of $\gamma$-subgradient projections between convex subsets given by Equations 5 and 6.

$$
\| y - H \beta \|_2 + \| \beta \|_2 < \epsilon_0
$$

$$
\| \beta \|_1 < \epsilon
$$

5.1.5. ABCS

Approximate Bayesian Compressive Sensing method [7] explores the use of a semi-Gaussian (SG) prior, and solves for $\beta$ in a Bayesian framework, thus allowing the complete statistics of the estimate of $\beta$ to be obtained. The motivation for using a SG prior can be motivated by analyzing the characteristics of the SG constraint $\| \beta \|_q = (\sum_i |\beta_i|^q)^{1/q}$ and the laplacian constraint $\| \beta \|_1 = (\sum_i |\beta_i|)$. We can denote the SG density function as proportional to $p_{\text{semi-gauss}} \propto \exp(-\| \beta \|_q^2)$ and the laplacian density function proportional to $p_{\text{laplace}} \propto \exp(-\| \beta \|_1)$. When $\| \beta \|_1 < 1$, it is straightforward to see that $p_{\text{semi-gauss}} > p_{\text{laplace}}$. When $\| \beta \|_1 = 1$, the density functions are the same, and when $\| \beta \|_1 > 1$ then $p_{\text{semi-gauss}} < p_{\text{laplace}}$. Therefore the semi-Gaussian density is more concentrated than the laplacian density in the convex area inside $\| \beta \|_1 < 1$. Given the sparseness constraint $\| \beta \|_q$ as the fractional norm $q$ goes to 0, the density becomes concentrated at the coordinate axes and the problem of solving for

![Figure 2: Error for RR and SR methods for varied $H$.](image-url)
\( \beta \) becomes a non-convex optimization problem where the reconstructed signal has the least mean-squared-error (MSE). As stated above, the semi-Gaussian density has more concentration inside the region \( \| \beta \| < 1 \). Intuitively, we expect the solution using the semi-Gaussian prior to behave closer to the non-convex solution.

With this in mind, the ABCS method can be formulated as:

\[
\min_{\beta} \| y - H\beta \|_2 + \lambda_1 (\beta - \beta_0)^T P_0^{-1} (\beta - \beta_0) + \lambda_2 \| \beta \|_1^2
\]

Here \( \beta_0 \) and \( P_0 \) are initialized statistical moments utilized in the algorithm, that is \( \beta_0 = E[\beta] \) and \( P_0 = var(\beta) \). For \( \beta_0 = 0 \) and \( P_0 \) represented by a diagonalized constant variance \( c \), the term \((\beta - \beta_0)^T P_0^{-1} (\beta - \beta_0)\) reduces to an l2 norm constraint similar to Elastic Net, namely \( c \| \beta \|_2^2 \). In [7] we explored using an informative prior for \( P_0 \). Specifically, we choose to initialize a diagonal \( P_0 \) where the entries corresponding to a particular class are proportional to the GMM posterior for that class. The intuition behind this is that the larger the initial \( P_0 \), the more weight is given to examples in \( H \) belonging to that class in ABCS. This provided roughly a 2% improvement in accuracy on TIMIT compared to not using a prior. However, in their current implementation, LASSO, BCS and Elastic Net do not allow for us to introduce an informative prior. Therefore, to fairly compare ABCS to the other techniques, we consider the case of \( P_0 \) as a constant \( c \).

5.2. Results

5.2.1. Visualization of Sparsity

First, we analyze the difference in \( \beta \) coefficients for different sparseness methods. For a randomly selected classification frame \( y \) in TIMIT and an \( H \) of size 200, we solve Equation 2 for \( \beta \). Figure 3 plots the sorted 200 \( \beta \) coefficients for four different techniques employing different regularizations, namely Ridge Regression, Lasso, Elastic Net and ABCS.

The plot shows that the \( \beta \) coefficients for the RR method are the least sparse, as we would expect. In addition, the LASSO technique has the sparsest \( \beta \) values. The sparsity of the Elastic Net and ABCS techniques methods are in between RR and LASSO, with ABCS being more sparse than Elastic Net due to the Semi-Gaussian constraint in ABCS, which is more sparse than the \( l_1 \) constraint in the Elastic Net.

5.2.2. TIMIT Results

Table 1 shows the results comparing various sparseness methods on TIMIT for a size of \( H = 200 \). As one can see from the table, the three methods which combine a sparseness constraint with and \( l_2 \) norm, namely ABCS, Elastic Net and CSP, all achieve statistically the same accuracy. The two methods which use the \( l_1 \) norm, namely BCS and LASSO, have slightly lower accuracy, showing the decrease in accuracy when a high degree of sparseness is enforced. Thus, it appears that using a combination of a sparseness constraint on \( \beta \), coupled with an \( l_2 \) norm, does not force unnecessary sparseness and offers the best performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>74.40</td>
</tr>
<tr>
<td>BCS</td>
<td>73.58</td>
</tr>
<tr>
<td>Elastic Net</td>
<td>77.89</td>
</tr>
<tr>
<td>ABCS</td>
<td>77.80</td>
</tr>
<tr>
<td>CSP</td>
<td>77.55</td>
</tr>
</tbody>
</table>

Table 1: Accuracies for Different Sparseness Methods

6. Conclusions and Future Work

In the paper we analyzed the effect of sparseness in speech processing. First, we explored under what conditions an SR approach is preferred over a RR method. On the TIMIT task, we observed the sparseness helps to prevent overfitting and remove sensitivity to outliers, relative to a RR method. This is particularly useful as the size of the dictionary \( H \) grows. Secondly, we explored the behavior of different types of sparseness methods on TIMIT and observed that methods which provide a combination between \( l_1 \) and \( l_2 \) norm, namely Elastic Net, CSP and ABCS, offer the best performance on TIMIT.

7. References