Enhanced Sparse Imputation Techniques for a Robust Speech Recognition Front-End

Qun Feng Tan*, Panayiotis G. Georgiou, *Senior Member, IEEE,* and Shrikanth S. Narayanan, *Fellow, IEEE*
Signal Analysis and Interpretation Laboratory (SAIL)
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089, U.S.A.
qtan@usc.edu,{georgiou,shri}@sipi.usc.edu

***Abstract*—Missing Data Techniques (MDT) have been widely employed and shown to improve speech recognition results under noisy conditions. This paper presents a new technique which improves upon previously proposed sparse imputation techniques relying on the Least Absolute Shrinkage and Selection Operator (LASSO). LASSO is widely employed in compressive sensing problems. However, the problem with LASSO is that it does not satisfy oracle properties in the event of a highly collinear dictionary, which happens with features extracted from most speech corpora. When we say that a variable selection procedure satisfies the oracle properties, we mean that it enjoys the same performance as though the underlying true model is known. Through experiments on the Aurora 2.0 noisy spoken digits database, we demonstrate that the Least Angle Regression implementation of the Elastic Net (LARS-EN) algorithm is able to better exploit the properties of a collinear dictionary, and thus is significantly more robust in terms of basis selection when compared to LASSO on the continuous digit recognition task with estimated mask. In addition, we investigate the effects and benefits of a good measure of sparsity on speech recognition rates. In particular, we demonstrate that a good measure of sparsity greatly improves speech recognition rates, and that the LARS modification of LASSO and LARS-EN can be terminated early to achieve improved recognition results, even though the estimation error is increased.**

EDICS: SPE-ROBU

***Index Terms*—Automatic Speech Recognition, Robustness, Convex Optimization, Sparse Representation, Compressive Sensing, Missing Data Techniques**

I. Introduction

MISSING data/feature techniques (MDT) have been proposed for noisy signal conditions to compensate for unreliable components of features corrupted by noise. By missing data/feature, we mean problems that are made difficult by absence of portions of data which take on some known/hypothesized structure. Missing data techniques have been employed in statistics [1] long before its adoption into the speech processing field for *Automatic Speech Recognition* (ASR). In addition to speech processing, techniques for data imputation (i.e. filling in or substituting for missing data) have also been employed in many other areas for denoising noisy measurements. For example, in the field of genetics [2], the microarrays employed in measuring gene expressions often suffer from the problem of probe noise. Another example is in reconstruction of noisy images [3].

There have been a large number of works pertaining to the topic of MDT and imputation in the speech processing field. For example, in [4], [5], the authors have employed two different statistical methods to infer the unreliable speech data. The first is marginalization, where the likelihood of the incomplete data vector is computed. Using x_r and x_u to denote the reliable parts and the unreliable parts of the feature vector respectively, the method allows computation of $p(x_r|C)$ instead of $p(x_r, x_u|C)$, where C represents the states in a *Hidden Markov Model* (HMM). A further refinement of this marginalization technique is termed "bounded marginalization" where the integral of the probability density functions are done over a finite range rather than from $-\infty$ to $+\infty$. The second method is to compute the distribution of the unreliable segments of the feature vector instead of the likelihood of the data present. Experimental evaluation on the TIDigits corpus with non-stationary (car/helicopter/factory) noise corruption showed that with these proposed techniques, the performance is much better than the original performance before imputation. In particular, the performance of the bounded marginalization method outperforms that of the second method.

Sparse representation techniques have also been used in the realm of MDT, attempting data reconstruction under the assumption that the signal can be reconstructed by a sparse representation from a dictionary. Sparse representation techniques and Compressive Sensing Techniques [6] (where the dictionary obeys the restricted isometry hypothesis) have been used widely, applications including phonetic classification in Speech Processing [7], and also Image Processing and Medical Imaging [8]–[12]. Recently, Gemmeke et al [13], [14] and Börgstrom et al [15], [16] have proposed the use of L_1 optimization techniques for spectral imputation. By L_1 optimization techniques, we are referring to techniques which optimize some error function subject to constraints on the L_1 norm of the solution vector $\boldsymbol{a}$, defined as follows:

$$||\boldsymbol{a}||_1 = \sum_n |a_n| \tag{1}$$

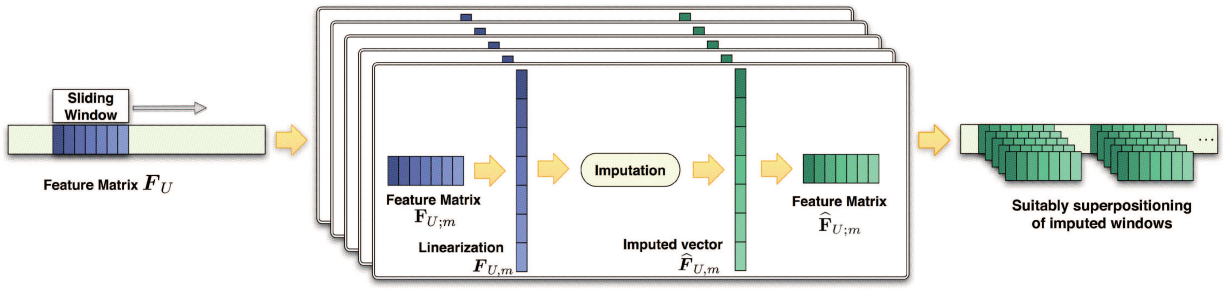

Fig. 1. Diagram of sparse imputation process

In particular, Gemmeke et al proposed an imputation framework based on a dictionary of exemplars, and refer to the process as "Sparse Imputation". Fig. 1 gives an illustration of the Sparse Imputation process. Both works have experimentally demonstrated the effectiveness of this technique when recovering missing speech components in adverse SNR conditions (SNR $\leq$ 0 dB). In [13], [14], the authors evaluated the imputation techniques on a time-normalized single digit recognition task. The formulation in [13], [14] assumes a well constructed dictionary for the sparse imputation process and it was found through experimentation that a dictionary with 4,000 exemplar spectrogram representations yielded the best performance with the LASSO algorithm in terms of speed and accuracy for their dataset. The LASSO algorithm is essentially a variable selection procedure which imposes a constraint on the L_1 norm of the solution vector. Results in [14] have demonstrated considerable improvement of the Sparse Imputation technique with well constructed dictionaries over classical imputation techniques like per-Gaussian-conditioned imputation and cluster-based imputation. An attempt to extend this system on the continuous digit task using LASSO has been reported in [17]. A further extension of the system to Large Vocabulary Continuous Speech Recognition (LVCSR) has also been explored in [18].

One of the goals of this paper is to investigate a solution to better exploit the properties of collinear dictionaries of exemplars in the sparse imputation setting for the continuous digits recognition task. We typically desire a dictionary which is less collinear, by which we are referring to a dictionary $\mathbf{B}$ where the value

$$C = \max_{1 \leq i,j \leq \dim(\mathbf{B},2)} |\mathbf{G}(i,j)| \text{ where } \mathbf{G} = \mathbf{B}^T\mathbf{B} \qquad (2)$$

is small [19], which essentially means that the entries of the dictionary have a higher tendency to point in different directions. Here, dim($\mathbf{B}$,2) refers to the number of columns in $\mathbf{B}$. In particular, in the event of a collinear dictionary, some variable selection procedures such as LASSO do not satisfy oracle properties, meaning that they do not identify the correct subset of predictors to model the observation, and they do not have an optimal estimation rate [20].

The two main algorithms we will investigate in this paper are LASSO [21] and the Least Angle Regression implementation of the Elastic Net (LARS-EN) [22] which is essentially an enhanced version of LASSO and *Ordinary Least Squares* (OLS). The reason LASSO is chosen as our baseline is because it has been demonstrated by [17] to be an efficient algorithm in the Sparse Imputation framework. In addition, LASSO offers a LARS modification which greatly accelerates its implementation. LARS-EN is chosen because it is theoretically proven to better exploit the property of collinear dictionary compared to LASSO, and offers the LARS modification which allows for fast execution [22].

We demonstrate experimentally that by better exploiting the properties of a collinear dictionary, we can expect to enjoy better speech recognition rates. We also provide a study of how different degrees of sparsity will affect speech recognition rates and why a good measure of sparsity is needed for optimal speech recognition results. We will use the implementation details of both algorithms to explain why a good measure of sparsity is necessary for optimal speech recognition rates. We demonstrate LARS-EN to be a significant improvement over LASSO for the continuous digit recognition task with estimated masks. We also supplement the results of evaluation with some popular regularization techniques for completeness. In this paper, like many others in the related literature, we will adopt the Aurora 2.0 noisy digits database for evaluation. The algorithms we introduce are incorporated into the speech recognition front-end, and the denoised/imputed version of the speech features are in turn used for speech recognition. We will be working with the standard *Mel-Frequency Cepstral Coefficient* (MFCC) front-end in contrast to [13], [14], which use PROSPECT features [23] for recognition. We also evaluate our methods on dictionaries of multiple sizes in contrast to [13], [14]. In practical scenarios, it will be difficult to tune for dictionary sizes, and the basis selection technique employed should ideally be able to select the appropriate sparse basis representation regardless of the structure of the dictionary. We will show that LARS-EN with small dictionary sizes outperforms LARS-LASSO with larger dictionaries for our digit recognition task.

The organization of the paper is as follows: Section II details the framework, algorithm, and justification as to the choices of the particular algorithms. Section III provides a description of our experimental setup and the experimental results, as well as

a discussion of our parameter choices, and the results. Section V concludes with possible extensions of this work.

II. Methodology

A. Construction of Representation of Test Utterances

We first need to construct a signal observation representation of the test spoken utterance U. In this work, we use frame-level spectral representations of speech rather than time-domain representations. The approaches in [13], [14] considered a fixed length vector representation for each digit. This is done by converting the acoustic feature representation to a time-normalized representation with a fixed number of acoustic feature frames. That reduces the digit recognition into a classification task since the digit boundaries are assumed to be known. Working with the assumption that the digit boundaries are known is non-trivial in practical settings.

In this paper, we consider the continuous digit recognition scenario like in [17]. Let the total number of frames for utterance U be denoted T_U. Let the feature vector corresponding to frame i of digit utterance U be denoted $\boldsymbol{f}_{U,i}$. The feature vector $\boldsymbol{f}_{U,i}$ contains N_B (number of frequency bands) spectral coefficients corresponding to frame i of utterance U. Let $\mathbf{F}_U$ be a $N_B \times T_U$ matrix defined as follows:

$$\mathbf{F}_U = \left(\boldsymbol{f}_{U,1}\ \boldsymbol{f}_{U,2}\ \cdots\ \boldsymbol{f}_{U,T_U}\right) \tag{3}$$

We now consider a sliding window extraction of the data in this matrix representation $\mathbf{F}_U$. Define a sliding matrix which has dimensions $N_B \times T_W$, T_W representing the duration of the sliding matrix. We also define a window shift parameter T_{WS}, which represents the number of frames by which we shift the sliding matrix.

Through this we obtain a total of $\lceil {}^{(T_U - T_W)}/_{T_{WS}} \rceil + 1$ matrices of feature vectors. For a more efficient implementation of the window extraction algorithm, we zero-pad $\mathbf{F_U}$ to be a $N_B \times k$ matrix where $k = \lceil {}^{(T_U - T_W)}/_{T_{WS}} \rceil \times T_{WS} + T_W$. Let us denote the m-th window corresponding to utterance U to be $\mathbf{F}_{U;m} = \left(\boldsymbol{f}_{U,m,1}\ \cdots\ \boldsymbol{f}_{U,m,T_W}\right)$. Let us denote the linearization of the matrix $\mathbf{F}_{U;m}$ to be the following:

$$\boldsymbol{F}_{U,m} = \begin{pmatrix} \boldsymbol{f}_{U,m,1} \\ \boldsymbol{f}_{U,m,2} \\ \cdots \\ \boldsymbol{f}_{U,m,T_W} \end{pmatrix} \tag{4}$$

Now, we make the assumption that we can write $\boldsymbol{F}_{U,m}$ as $\boldsymbol{F}_{U,m} = \mathbf{B}\boldsymbol{a}_m$, where $\boldsymbol{F}_{U,m}$ is the observation (feature vector), $\mathbf{B}$ is a dictionary of exemplars, $\boldsymbol{a}_m$ is a vector of weights. We are assuming that each test segment can be written as a linear combination of vectors from the dictionary. This is a reasonable assumption to make and follows the approaches in [10], [13], [14], [24] and many other signal processing applications where a regularized regression setting is desired for denoising. Also, the spectral representations for different realizations of the same word have energy concentrations in similar regions in the time-frequency domain, giving us a reason for using this linear representation.

Thus, we will have the following linear representations from our windows:

$$\boldsymbol{F}_{U,m} = \mathbf{B}\boldsymbol{a}_m, \quad m = 1, ..., \left\lceil \frac{T_U - T_W}{T_{WS}} \right\rceil + 1 \tag{5}$$

After the sparse imputation process, we need to reconstruct an imputed representation of the original sliding matrix. Define a counter matrix of dimension $N_B \times k$ where $k = \lceil {}^{(T_U - T_W)}/_{T_{WS}} \rceil \times T_{WS} + T_W$. This counter matrix counts the number of times each entry in the matrix $\mathbf{F}_{U;m}$ is imputed due to overlapping windows. Formation of the final imputed matrix will involve first reshaping $\widehat{\boldsymbol{F}}_{U,m}$ (the solution to optimizing Equation 5) back to dimensions $N_B \times k$, adding all the resulting reshaped frames together, and then doing element-wise division by the entries of the counter matrix. This is in effect amounts to averaging the contributions of individual imputations coming from multiple windows. To simplify the notation we will omit the subscript m in the remainder of the paper when dealing with the m-th sliding matrix.

Let us denote the number of utterances in the training data to be used in our dictionary by N_{train}. We then form a dictionary $\mathbf{B} = [\boldsymbol{B}_1\ \boldsymbol{B}_2\ ...\ \boldsymbol{B}_{N_{\text{train}}}]$ which consists of segments of clean spectral shapes. This will be our overcomplete dictionary of exemplar spectral segments. We now describe the procedure by which we obtain $\boldsymbol{B}_i$ for $i = 1, ..., N_{\text{train}}$. To motivate our dictionary choice, we will treat each digit as a recognition unit, and hence we will be putting exemplars of the clean spectral shapes of each recognition unit as entries of our dictionary. We thus only consider the single digit files in our training data for formation of our dictionary, since we will have whole digit utterances without having to do any forced alignment. We then extract the $N_B \times T_U$ matrix containing the spectral coefficients corresponding to those digits. After extraction, a simple time normalization is performed by interpolating the T_U frames of the extracted spectral features of the single digit files to T_W frames. This is done by cubic spline interpolation [25] which retains the spectral shapes of the coefficients fairly well. We then linearize the interpolated matrix to a $N_B \cdot T_W \times 1$ vector which goes into each column of our dictionary $\mathbf{B}$. Note that our dictionary construction retains boundary information to a great extent, which turns out to be instrumental in improving recognition rates. Section III-E1 will provide experimental evidence to substantiate our dictionary choice over randomly selected fixed-length exemplars for the continuous digit recognition task.

B. Signal Reliability Masks

Most works in speech processing MDT [4], [5], [13], [14] define some sort of signal reliability mask for the mel-frequency log-energy coefficients (the popularly used signal representation), which is a matrix the size of the original feature vectors with entries containing 1 to mean that the feature component is reliable and 0 to mean domination by noise. Basically, the mask is defined as:

$$M(k,t) = \begin{cases} 1 & \text{if } \frac{S(k,t)}{N(k,t)} > \lambda_{\text{SNR}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where S and N stands for the signal and noise respectively. k is the index of the frequency bands and t the index of the time frames.

If the signal-to-noise (SNR) ratio is above a certain threshold we deem appropriate, we regard the component to be reliable. However, if the SNR is below our threshold, this means the component is unreliable and will be replaced with the corresponding imputed version we have from our imputation algorithms.

An oracle mask is a signal reliability mask computed with perfect knowledge of what the noise signal is and what the underlying clean signal is. An estimated mask relaxes the assumption that we have oracle knowledge of the underlying noise characteristics by estimating the noise characteristics from the noisy speech signal itself.

There have been works to estimate masks from just the observed noisy speech data, like in [26], where MDT techniques were evaluated on a variety of masks such as Discrete SNR Masks, Soft SNR Masks, and combined Harmonicity and SNR masks. Another good overview on the topic of mask estimation is given in [27].

We adopt the estimated mask described in [28] rather than using oracle masks as described in [4]. Essentially, we get a local estimate of the SNR by averaging the first 10 frames of the spectral features of the utterance, which contains information preceding the voicing of the digits. This provides a reasonable estimate of the noise, under the condition that the noise is stationary [28], which will be assumed here. An estimate of the clean digit utterance is obtained by subtraction of the noise estimate from the noisy digit utterance.

Now that we have a mask giving us an indication of the reliable/unreliable components in the spectrum, we are able to make modifications to our dictionary $\mathbf{B}$. The main idea is that we will be only including vectors in the dictionary which correspond to reliable components for our imputation process. Let N_r be the number of reliable components in $\mathbf{F}_U$. We define the matrix R to be a $N_r \times N_B \cdot T_W$ matrix containing 0's and 1's which extracts the rows of $\mathbf{B}$ which correspond to the reliable components of $\mathbf{F}_U$ as defined by our estimated mask. Thus, we have

$$\begin{aligned} \mathbf{B}_r &= \mathbf{RB} \\ \mathbf{F}_{Ur} &= \mathbf{RF}_U \\ \mathbf{F}_{Ur} &\approx \mathbf{B}_r \boldsymbol{a} \end{aligned} \tag{7}$$

where $\mathbf{B}_r$ is the new dictionary we use for our algorithms and $\mathbf{F}_{Ur}$ are the reliable components in $\mathbf{F}_U$. For the reconstruction, we will simply impute the components which are defined as unreliable by our SNR mask.

C. Noise Model

When noise is added to the speech signal, the spectral coefficients will be perturbed. Let us represent this perturbation in our model as

$$\boldsymbol{F}_{U_r} = \mathbf{B}_r \boldsymbol{a} + \boldsymbol{z} \tag{8}$$

where $\boldsymbol{z}$ is a $N_r \times 1$ dimension noise vector. Note that even though we have been dealing with what is in principle a noiseless signal corresponding to the reliable parts of the data, in practice there will still be a noise component associated with it, which we attempt to capture by the vector $\boldsymbol{z}$.

We assume that we are dealing with additive noise in the time domain, and hence we have $O(t) = S(t) + N(t)$, where $O(t)$ is the observed speech signal, $S(t)$ is the original clean speech signal, $N(t)$ is the noise signal, and t refers to the time frame. To compute the spectral coefficients using the notation in [29], we do pre-emphasis, frame blocking, windowing (Hamming), and the *Short-Time Discrete Fourier Transform* (stDFT) to give $S(k,l)$ where k refers to the frequency band index and l refers to the length of the window used for the stDFT. Since the stDFT is linear, we will analogously have the additive noise become additive spectral noise $N(k,l)$. Taking the logarithm, we will have $\log|S(k,l) + N(k,l)|$, which we can then write as $\log\left|S(k,l)\left(1+\frac{N(k,l)}{S(k,l)}\right)\right| = \log|S(k,l)| + \log\left|\left(1+\frac{N(k,l)}{S(k,l)}\right)\right|$. We can hence calculate the output $Y(i)$ of the i-th critical band filter by:

$$\sum_{k=0}^{\frac{N'}{2}} \left(\log|S(k,l)| + \log\left|\left(1+\frac{N(k,l)}{S(k,l)}\right)\right|\right) H_i\left(k\frac{2\pi}{N'}\right) \tag{9}$$

H_i is the impulse response of the i-th critical band filter and N' is the number of points for computing the stDFT. Ideally, a high SNR will mean that $\frac{N(k,l)}{S(k,l)}$ is close to zero, and so the term $\log\left|\left(1+\frac{N(k,l)}{S(k,l)}\right)\right| H_i\left(k\frac{2\pi}{N'}\right)$ in Equation (9) will be approximately zero, meaning that the observed spectral component will be close to the true value. However, in reality, there will still be a mismatch between the estimated reliable components and their true values; thus we can attempt to model the term $\log\left|\left(1+\frac{N(k,l)}{S(k,l)}\right)\right| H_i\left(k\frac{2\pi}{N'}\right)$ by some appropriate noise model in our optimization problem. Hence, we see that spectral imputation is closely equivalent to imputation of the output features.

D. Bounded Optimization

Bounded optimization refers to solving the optimization problem such that the optimized value is less than or equal to the original value. Unbounded optimization means that this constraint is ignored. Since we are approximating the additive noise in the time domain by additive noise in the spectral domain also, the imputed values should technically be less than the original noisy version. However, our optimization problems are generally unbounded; hence this constraint is not guaranteed. To circumvent this problem, as in most works in MDT, we simply opted to impute only if the specific component to be imputed has an inferred value which is less than the original noisy component. In general, this simple rule resulted in better recognition accuracies compared to those of the unbounded situation by our preliminary experiments. This phenomena has been observed in [14] as well.

E. Formulation of optimization problem

If we consider a regularized least-squares approach for the spectral coefficients denoising, the vector $\boldsymbol{a}$ is assumed

distributed according to a Gaussian distribution [30]. By a similar token, if we consider a regularized L_1 approach, $\boldsymbol{a}$ is assumed Laplacian. To ensure maximal sparsity, we ideally like to solve the L_0 optimization problem. For our problem setup, this is equivalent to solving the optimization problem of the form:

$$\min_{\boldsymbol{a}} \|\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}\|_2 + \lambda \|\boldsymbol{a}\|_0 \tag{10}$$

Here, the parameter λ controls the sparsity of the vector $\boldsymbol{a}$; specifically, when we increase the value of λ, $\boldsymbol{a}$ will become more sparse.

It is a well-known fact that optimizing equation (10) is an NP-hard problem, since it involves searching through $C_k^{N_{\text{train}}}$ least-squares problems, where k is the degree of sparsity desired. This can be computationally expensive for our problem at hand since N_{train} could potentially be large. There have been alternatives proposed which try to get around this problem while still maintaining a good penalty curve approximation to the L_0 solution.

F. Baseline - The LASSO solution

Schemes presented in [13]–[16] have employed the classical L_1 solution for sparse imputation borrowed from compressive sensing [31], [32], and have demonstrated its efficiency in the sparse imputation framework. Furthurmore, the penalty curve for L_1 optimization emulates that of the L_0 solution fairly closely (See Fig. 2). We will likewise use the classical L_1 solution as our baseline. When applied to our problem setup, we can represent it as the following convex optimization problem:

$$\min_{\boldsymbol{a}} \|\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}\|_2 + \lambda \|\boldsymbol{a}\|_1 \tag{11}$$

Note that Equation (11) can be equivalently formulated as:

$$\begin{aligned} &\min_{\boldsymbol{a}} \|\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}\|_2 \\ &\text{subject to } \|\boldsymbol{a}\|_1 \leq t \end{aligned} \tag{12}$$

Here t is the shrinkage parameter, which is inversely related to λ. As t gets smaller, the weight vector $\boldsymbol{a}$ will become more sparse. Note that it is easy to see that equations (11) and (12) are equivalent by the *Karush-Kuhn-Tucker* (KKT) conditions [32].

There are several efficient algorithms proposed for the solution of (11) or (12). [21] proposed the "*Least Absolute Shrinkage and Selection Operator*" (LASSO) which involves a series of quadratic programs. In fact, Equation (12) is equivalent to solving a least-squares with $2^{N_{\text{train}}}$ different inequality constraints corresponding to the signs of the components of a:

$$a_i \leq t_i \text{ or } \ a_i \geq -t_i \text{ for } i = 1, ..., N_{\text{train}} \tag{13}$$

What Tibshirani [21] proposed is that, instead of solving all the inequality constraints at once, we can progressively incorporate the inequality constraints while seeking a solution which still satisfies the KKT conditions. Essentially, the iterative algorithm starts out with just the sign of the least-squares solution in its constraint set. If the next iteration solves the problem, the algorithm is terminated. Otherwise, the violated constraint is added to the constraint set and the algorithm continues.

Another solution proposed by Efron et al for the LASSO algorithm is the *Least Angle Regression* (LARS) modification for LASSO (LARS-LASSO) [33]. One of the important results in the paper is proving that the LARS algorithm yields all LASSO solutions. The LARS algorithm is a much faster algorithm compared to Tibshirani's original proposal in [21]. It starts by setting all coefficients to zero and then finding the highest direction of correlation with the response vector. It then takes the largest step possible in that direction until some other predictor has as much correlation with the current residual. LARS then continues in the direction equiangular between the two predictors, and this procedure is repeated. This is actually an important property which we will capitalize upon to control sparsity, and the experimental justification for this will be presented in Section III.

Another related work regarding the L_1 optimization is Basis-Pursuit [34]. We use the LARS-LASSO algorithm as our baseline.

G. Drawbacks of the classical LASSO solution for our recognition task and dataset

There have been several studies on the asymptotics and oracle properties of LASSO such as in [20], [35]–[37].

In [20], [37], it has been independently demonstrated that LASSO does not satisfy oracle properties under certain circumstances. Let us define $\widehat{\boldsymbol{a}}$ to be the estimate returned by a variable selection procedure. We define the oracle properties of a variable selection procedure as follows [20]:

- Identify the right subset of predictors to model the observation
- Have an optimal estimation rate given by: $\sqrt{N_r}(\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{a}}^*) \rightarrow N(\mathbf{0}, \boldsymbol{\Sigma})$. $\widehat{\boldsymbol{a}}^*$ is the estimate where $E[F_{U_r}|\mathbf{B_r}] = \mathbf{B_r}\widehat{\boldsymbol{a}}^*$ and $\boldsymbol{\Sigma}$ is the covariance matrix given that the oracle subset of predictors is known.

Without loss of generality, assume that the first p entries of $\widehat{\boldsymbol{a}}$ are non zero, $p < N_{\text{train}}$, where here $\widehat{\boldsymbol{a}}$ is the solution to the optimization problem (11). Otherwise, we can easily reorder the columns of $\mathbf{B}_r = \left[\boldsymbol{B}_{r_1}...\boldsymbol{B}_{r_{N_{\text{train}}}}\right]$ to match this assumption. Also assume the rest of the entries $\widehat{\boldsymbol{a}}_i = 0$, where $i > p$.

Define the covariance matrix $\mathbf{C}$ to be:

$$\mathbf{C} = \frac{1}{N_r}\mathbf{B}_r^T\mathbf{B}_r = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix} \tag{14}$$

where $\mathbf{C}$ is a positive definite matrix. We set

$$\begin{aligned}
\mathbf{C}_{11} &= \frac{1}{N_r}\left[\boldsymbol{B}_{r_1}\ ...\ \boldsymbol{B}_{r_p}\right]^T\left[\boldsymbol{B}_{r_1}\ ...\ \boldsymbol{B}_{r_p}\right] \\
\mathbf{C}_{22} &= \frac{1}{N_r}\left[\boldsymbol{B}_{r_{p+1}}\ ...\ \boldsymbol{B}_{r_{N_{\text{train}}}}\right]^T\left[\boldsymbol{B}_{r_{p+1}}\ ...\ \boldsymbol{B}_{r_{N_{\text{train}}}}\right] \\
\mathbf{C}_{12} &= \frac{1}{N_r}\left[\boldsymbol{B}_{r_1}\ ...\ \boldsymbol{B}_{r_p}\right]^T\left[\boldsymbol{B}_{r_{p+1}}\ ...\ \boldsymbol{B}_{r_{N_{\text{train}}}}\right] \\
\mathbf{C}_{21} &= \frac{1}{N_r}\left[\boldsymbol{B}_{r_{p+1}}\ ...\ \boldsymbol{B}_{r_{N_{\text{train}}}}\right]^T\left[\boldsymbol{B}_{r_1}\ ...\ \boldsymbol{B}_{r_p}\right] \qquad (15)
\end{aligned}$$

We say that the estimator $\widehat{\boldsymbol{a}}$ is sign consistent if and only if

$$P(\text{sign}(\boldsymbol{a}) = \text{sign}(\widehat{\boldsymbol{a}})) \rightarrow 1 \text{ as } N_r \rightarrow \infty \qquad (16)$$

Sign consistency is needed for the LASSO estimate to match the true model.

It is proven in [20], [37] that LASSO is sign consistent only if $|\mathbf{C}_{21}\mathbf{C}_{11}^{-1}\text{sign}([\widehat{\boldsymbol{a}}_1\ ...\ \widehat{\boldsymbol{a}}_p]^T)| \leq \mathbf{1} - \eta$ (Strong Irrepresentable Condition). $\mathbf{1}$ is a $p \times 1$ dimensional vector of ones. However, it is easy to see that this condition is easily violated when the columns of $\mathbf{B}_r$ are highly collinear or correlated.

In our case, we can expect that the spectral profiles (coefficients) for the same digits to be similar. Thus we can expect contiguous entries in the dictionary to be highly collinear. Moreover, in our specific overcomplete dictionary, there are only 11 distinct digits. Thus, we will have a highly coherent dictionary, potentially leading to problems when using LASSO.

H. Proposed solution for the continuous digits task

One of the possible alternatives for testing the suitability of the Gaussian noise model is the *Ordinary Least-Squares* (OLS) method. However, it is well known that OLS is generally inferior in terms of prediction and does not give parsimonious solutions [22]. Moreover, its penalty curve is less effective than LASSO when used as an approximation for the L_0 penalty curve, since OLS has a quadratic penalty curve. LASSO has a linear penalty curve, which emulates the L_0 penalty curve better than the quadratic one (See Fig. 2).

To circumvent the disadvantages of LASSO for our problem as outlined in Section II-G, and also that of OLS, several solutions have been proposed. They include the Elastic Net solution (a variation of regularized least-squares) [22], *Sparse Bayesian Learning* (SBL) [38], [39], Matching Pursuit [40] and Orthogonal Matching Pursuit [41].

SBL assumes a parametrized prior from the data using a Gamma distribution and by choice of this distribution, is shown to enjoy sparsity. While SBL is guaranteed to converge to an optimum since it operates with the *Expectation-Maximization* (EM) algorithm, it could potentially get stuck in a local minimum. In fact, our experiments with SBL also resulted in lower recognition accuracies as compared to LARS-LASSO for the continuous digits recognition task (See Appendix for results) for high λ_{SNR} thresholds. Bayesian compressive sensing techniques have, however, demonstrated success in the phonetic classification task [42].

We have additionally evaluated Matching Pursuit and Orthogonal Matching Pursuit which are popular L_0-approximation techniques, but both algorithms performed worse than LARS-LASSO in terms of recognition rates for our imputation task (See Appendix for results of our evaluation) .

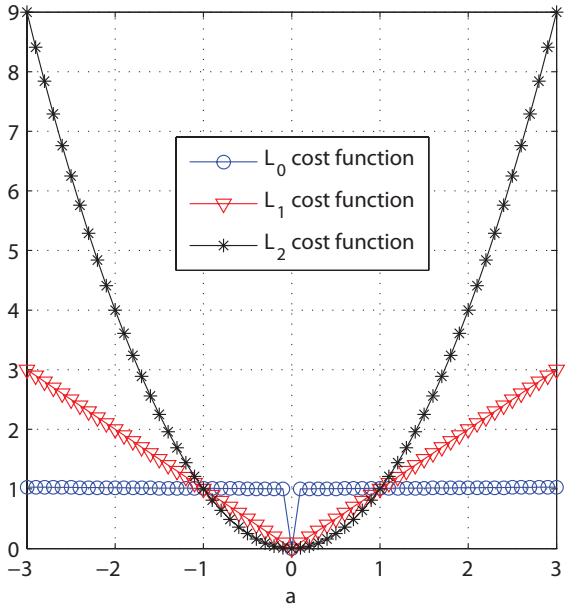

Fig. 2. Visualization of the L_0, L_1, and L_2 penalty functions. In the figure, we can see that the linear L_1 penalty function emulates the L_0 penalty function more closely than the quadratic L_2 penalty function.

The Elastic Net is our choice for the optimization task due to the advantages advocated by the authors in [22]. In particular, the Elastic Net encourages a "grouping effect" more strongly than LASSO, where highly correlated predictors tend to be selected or excluded together in a more efficient manner. The Elastic Net is also more effective as a variable selection procedure when we encounter a matrix where the number of columns is much greater than the number of rows, as with our current framework. Moreover, the Elastic Net can be viewed as a more general framework as an extension to LASSO, since LASSO is a special case of the Elastic Net when the coefficient for the L_2 regularization term is set to zero. Computationally, the Elastic Net offers a LARS modification which allows us to create the entire solution path with complexity comparable to that of a single OLS optimization, and thus is efficient compared to OLS.

The "naive" Elastic Net formulation is given by the following:

$$\underset{\boldsymbol{a}}{\text{argmin}}\ ||\boldsymbol{F}_{Ur} - \mathbf{B}_r\boldsymbol{a}||_2^2 + \lambda_1||\boldsymbol{a}||_1 + \lambda_2||\boldsymbol{a}||_2^2 \qquad (17)$$

Note that the formulation in Equation (17) is very similar to our original formulation in Equation (11), with an additional regularization term of the L_2 norm.

The reason why the formulation in Equation (17) is called "naive" is due to the fact that experimental evidence by the authors in [22] showed that it does not perform up to expectations unless it is close to the ridge regression or LASSO. The solution to this is scaling the solution of the "naive" Elastic Net as follows:

$$\boldsymbol{a}_{\text{Elastic Net}} = (1+\lambda_2) \times \boldsymbol{a}_{\text{"naive" Elastic Net}} \tag{18}$$

From this basic formulation, it is possible to prove that the Elastic Net overcomes some of the limitations of LASSO [22]; most importantly, it is able to better exploit the properties of highly collinear/correlated dictionary entries. Specifically, in the event of a group of highly correlated vectors, LASSO has a tendency to randomly pick one from the group without regard for which one is selected. However, the Elastic Net is demonstrated to be able to select these "grouped" variables more efficiently.

We will be using the Least Angle Regression implementation of the Elastic Net (LARS-EN), which implements the Elastic Net in an efficient manner as mentioned above. Further details of this implementation can be found in [22].

III. Description of dataset, experimental setup, and experimental results

A. Experimental Setup

1) Database: For our recognition system, we use all of the 8040 clean training files (containing single and continuous digit utterances) provided in the Aurora 2.0 database training set to train a continuous digit recognizer in HTK [43].

For the continuous digit recognition task, the Aurora database consists of test sets labeled N1, N2, N3 and N4 (corresponding to subway, babble, car and exhibition noise respectively) in the Test Set A subset. We used the N1 folder for tuning of our optimization parameters. For the test sets, we created two test sets as follows:

- TEST1: merging N1, N2, N3 and N4, giving us a total of 4004 files
- TEST2: merging N2, N3 and N4 (exclusion of the N1 folder), giving us a total of 3003 files

We evaluated our algorithms on different SNR conditions: SNR -5 dB, SNR 0 dB, SNR 5 dB and SNR 10 dB.

To form our dictionary $\mathbf{B}$, we get all the single digit audio files in the training data, which totals 2412 files, and interpolate them to T_W frames as described above. We then form $\mathbf{B}$ with each column representing the interpolated spectral components from the 2412 files, giving us a matrix $\mathbf{B}$ of dimensions $N_B.T_W \times 2412$. We also experiment with different dictionary sizes, namely with $N_{\text{train}} = 1000, 1500, 2000, 2412$.

2) ASR Features: We train the recognizer on MFCCs with the first and second derivatives, with 16 states total. We use 23 frequency bands ($N_B = 23$), a hamming window size of 25 ms, and a frame shift of 10 ms. For the delta and delta-delta coefficients, we set the DELWINDOW and the ACCWINDOW parameters in HTK to be both equal to 2 frames.

The feature extraction for the 23 spectral coefficients is done in MATLAB. We then optimize upon these spectral coefficients with the optimization algorithms described in the Section II-H. From the denoised spectral coefficients, we reconstruct (again using MATLAB) the 13 MFCC coefficients with the first and second derivatives, which are then fed to the HTK continuous digit recognizer that we have trained.

3) Algorithm Implementation details: Both our optimization algorithms are implemented using MATLAB. The LARS-LASSO baseline was implemented using Sparselab available at http://sparselab.stanford.edu which provides a MATLAB routine called SolveLasso to solve the LASSO formulation using the LARS modification.

B. Tuning for parameters

TABLE I
PARAMETER VARIATION/TUNING FOR SNR 5 dB DATASET FOR THE AURORA 2.0 DATABASE ON CONTENTS OF TESTA/N1. THE LAST COLUMN INDICATES WHETHER THE IMPROVEMENT OVER LARS-LASSO AT THE SAME VALUE OF λ_{SNR} IS SIGNIFICANT WITH THE DIFFERENCE OF PROPORTIONS TEST AT 95% CONFIDENCE LEVEL. THE BEST PERFORMING RESULT OF EACH ALGORITHM IS IN BOLD

Alg	λ_{SNR} in dB	Degree of Sparsity	Iterations	Accuracy (%)	Significant ?
Unimputed	NA	NA	NA	52.48	NA
LARS-LASSO	-13.98	NA	5	22.45	NA
LARS-LASSO	3	NA	5	38.50	NA
LARS-LASSO	7	NA	5	54.66	NA
LARS-LASSO	7	NA	10	60.49	NA
LARS-LASSO	10	NA	5	58.92	NA
LARS-LASSO	10	NA	10	63.64	NA
LARS-LASSO	10	NA	20	63.52	NA
LARS-LASSO	20	NA	10	**65.88**	NA
LARS-LASSO	20	NA	15	63.19	NA
LARS-LASSO	30	NA	10	58.59	NA
LARS-EN	3	50	NA	53.65	YES
LARS-EN	7	50	NA	65.32	YES
LARS-EN	10	5	NA	60.38	NO
LARS-EN	10	10	NA	63.19	NO
LARS-EN	10	15	NA	66.33	YES
LARS-EN	10	30	NA	67.23	YES
LARS-EN	10	50	NA	67.12	YES
LARS-EN	20	30	NA	71.72	YES
LARS-EN	20	50	NA	**71.83**	YES
LARS-EN	20	70	NA	71.83	YES
LARS-EN	30	50	NA	69.25	YES

For parameter tuning, we used a smaller subset of the test files we have. We took 1001 test files from the test set testa/N1 as our tuning set. The tuning results are presented in Table I.

As evident from Table I, we used the SNR 5 dataset to tune for a suitable reliability threshold λ_{SNR}. Since we are using an estimated mask, we will want more accurate components to enter our optimization matrix. Thus we need to set our confidence level to be sufficiently high to eliminate bad estimates. At the same time, if we set the threshold of λ_{SNR} to be too high, too few components will enter our optimization matrix and we will have an ill-defined optimization problem. Thus, it is important to strike a balance between these two factors. We decided upon a λ_{SNR} threshold of 20 dB after some experimentation (see results in Table I).

Initial experimentation with several window lengths showed that $T_W = 35$ (which is also the average digit duration in the training set) is a good window length to choose for our particular database. For the frame shift parameter, we experimented with several values using LARS-LASSO as the tuning algorithm. We find that $T_{WS} = 10$ provided the best results for the SNR 5 set. Note that for our tuning set using LARS-LASSO, $T_{WS} = 1$ gave a recognition rate of 63.24%,

$T_{WS} = 5$ a recognition rate of 65.04%, $T_{WS} = 10$ a recognition rate of 65.88%, and $T_{WS} = 15$ a recognition rate of 45.38%. The reason behind the differences with [17] is due to a different dictionary construction. Thus, T_{WS} has a different optimal point to ensure optimal superpositioning of the imputed results for the best speech recognition rates.

As for the algorithmic parameters, we decided upon 10 iterations of LARS-LASSO and a sparsity degree of 50 for the LARS-EN by experimentation with our tuning set (see Table I). Note that the number of iterations and sparsity degree is in fact related; the more iterations of the LARS algorithm we take, the less sparse our solution vector will be (See Fig. 3).

C. Experimental Results for Continuous Digit Recognition Task

For the continuous digit task, we evaluated LARS-EN and LARS-LASSO (baseline).

TABLE II
RECOGNITION RESULTS FOR DIFFERENT NOISE CORRUPTION VALUES RANGING FROM -5 dB, 0 dB, 5 dB AND 10 dB FOR TEST1. THE LAST COLUMN INDICATES WHETHER THE IMPROVEMENT OVER LARS-LASSO IS SIGNIFICANT WITH THE DIFFERENCE OF PROPORTIONS TEST AT 95% CONFIDENCE LEVEL

Alg	Degree of Sparsity	Iterations	Accuracy (%)	Significant ?
SNR = -5 dB				
Unimputed	NA	NA	7.89	NA
LARS-LASSO	NA	10	29.74	NA
LARS-EN	50	NA	31.04	YES
SNR = 0 dB				
Unimputed	NA	NA	18.59	NA
LARS-LASSO	NA	10	44.54	NA
LARS-EN	50	NA	46.83	YES
SNR = 5 dB				
Unimputed	NA	NA	43.82	NA
LARS-LASSO	NA	10	64.28	NA
LARS-EN	50	NA	66.36	YES
SNR = 10 dB				
Unimputed	NA	NA	68.89	NA
LASSO	NA	10	79.17	NA
LARS-EN	50	NA	82.16	YES

We make the following observations from our experimental results given in Tables I,II, and III:

1) *LARS-EN consistently out-performs LARS-LASSO in terms of recognition accuracy.* This can be explained by LARS-EN being more adept at handling a collinear dictionary as compared to LARS-LASSO.
2) *As we start from a degree of sparsity of zero and continue increasing the value, the recognition rate increases. However, there is a point of saturation; any further increase in the number of non-zero components leads to degradation of the speech-recognition performance.* As we start increasing the number of non-zero entries in $\boldsymbol{a}$, we keep getting a better representation of the observation vector $\boldsymbol{F}_{Ur}$. However, after a point, the representation becomes poor due mainly to the fact the LARS modification is essentially a greedy approach. As characteristic to most greedy algorithms (like Forward Selection/Forward Stagewise), each movement is in the promising direction, and it is highly likely that as more iterations are taken, some important covariates are missed, resulting in errors in speech recognition. Each step forward in a different direction corresponds to an increase in one degree of sparsity. This testifies to the fact that a good sparsity measure helps in improving recognition rates.
   Another reason why sparsity is needed is due to the fact that it helps prevent overfitting by ensuring that the more relevant components are chosen.
3) *As we increase λ_{SNR}, the recognition improves until a point where it saturates.* See Section III-B for further details.
4) *The recognition rates generally do not depend on the magnitude of the estimation error $||\boldsymbol{F}_{Ur} - \mathbf{B}_r\boldsymbol{a}||_2$, but rather on the quality of covariates selected.* This is in fact an interesting and important observation. For example, for LARS-LASSO, we see that 10 iterations of the algorithm give a better performance than 20 iterations of the algorithm from Table I. In fact, a small number of iterations gives a sparser vector a compared to a larger number iterations of the algorithm but at the expense of a higher error in $||\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}||_2$ by nature of the LARS implementation. The values of $||\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}||_2$ for the Sparselab implementation of LARS-LASSO can be verified by setting the Verbose option to be True. Fig. 3 gives examples of stem plots of the solution vectors for a specific optimization. Fig. 4 shows a plot of the recognition rate vs. the average estimation error across all the optimizations of our tuning set. This can be explained by the fact that ASR accuracy is more determined by the relevant covariates that the regression technique selects rather than the absolute error that the optimization problem seeks to minimize. In fact, it is apparent from our experiments that if bad covariates are

TABLE III
RECOGNITION RESULTS FOR DIFFERENT NOISE CORRUPTION VALUES RANGING FROM -5 dB, 0 dB, 5 dB AND 10 dB FOR TEST2. THE LAST COLUMN INDICATES WHETHER THE IMPROVEMENT OVER LARS-LASSO IS SIGNIFICANT WITH THE DIFFERENCE OF PROPORTIONS TEST AT 90% CONFIDENCE LEVEL

Alg	Degree of Sparsity	Iterations	Accuracy (%)	Significant ?
SNR = -5 dB				
Unimputed	NA	NA	7.95	NA
LARS-LASSO	NA	10	29.33	NA
LARS-EN	50	NA	30.61	YES
SNR = 0 dB				
Unimputed	NA	NA	17.59	NA
LARS-LASSO	NA	10	43.41	NA
LARS-EN	50	NA	45.35	YES
SNR = 5 dB				
Unimputed	NA	NA	40.95	NA
LARS-LASSO	NA	10	63.69	NA
LARS-EN	50	NA	64.35	NO
SNR = 10 dB				
Unimputed	NA	NA	65.77	NA
LASSO	NA	10	78.31	NA
LARS-EN	50	NA	80.92	YES

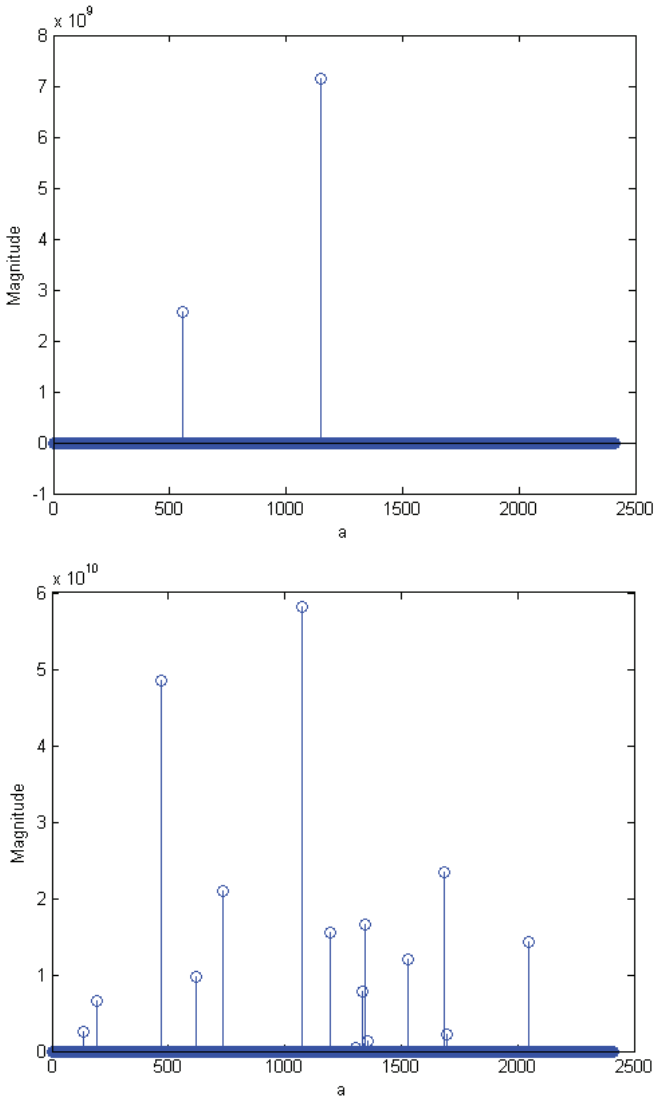

Fig. 3. The stem plot on the top shows the weight vector for 5 iterations of LARS-LASSO, and the one on the bottom shows the weight vector for 20 iterations of LARS-LASSO for a specific optimization problem. While 5 iterations of LARS-LASSO give a sparser representation than 20 iterations of LARS-LASSO from the diagram, 5 iterations of LARS-LASSO will have a higher error in $||\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}||_2$ compared to 20 iterations due to the nature of the LARS implementation.

included, even if the error in estimation is much lower, the recognition rates suffer.

Another connection to sparsity is to relate this observation to Equation 11. Note that an increase in the magnitude of λ implies an increase in sparsity. When λ becomes big, the role that $||\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}||_2$ plays in the optimization problem decreases. This further reinforces the fact that the appropriate degree of sparsity helps in improving recognition rates rather than the low magnitude in the estimation error $||\mathbf{B}_r\boldsymbol{a} - \boldsymbol{F}_{Ur}||_2$.

D. Insight into the mechanism behind the imputation process

Fig. 5 shows the spectral plots of one particular utterance. As we can observe, the imputed spectral plots bear much closer resemblance to the clean signal than the noisy one, with much of the noise artifacts removed. In particular, we can distinctly observe that the imputed result with LARS-EN bears a closer resemblance to the clean signal than LARS-LASSO. This further testifies to the robustness of LARS-EN given our experimental setup/conditions.

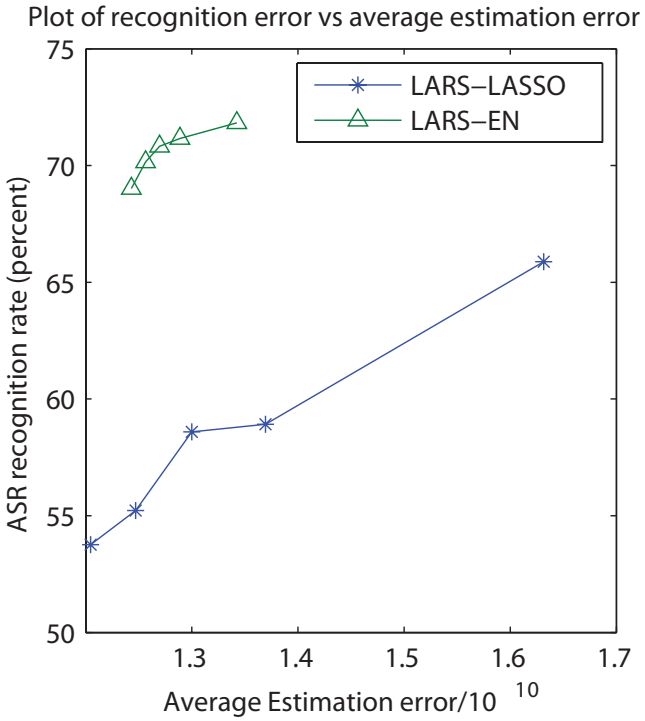

Fig. 4. Plot of ASR recognition error vs average estimation error rate (average across all optimization problems) for the SNR 5 corruption level. We can see that a diminishing estimation error rate results in a much worse performance in terms of ASR recognition rate for both LARS-LASSO and LARS-EN. The highest points in both graphs correspond to the best performance that was evaluated with the algorithms for the tuning set.

In fact, the regularization techniques, when deployed in the spectral domain, can be viewed as spectral denoising. In each frame of imputation, we are essentially doing some form of spectral profile identification. To further reinforce this intuition, the first step of the LARS-LASSO involves finding the projection of the observation vector onto the dictionary $\mathbf{B}_r$. Thus, what we are doing is essentially a form of spectral profile identification, taking into account a noise model. The sliding window framework simply reconciles all the possible predictions by careful superposition of the predictions and then averaging them.

E. Investigation of various dictionary structures

1) Whole digits vs. randomly selected fixed length exemplars: For $\lambda_{\text{SNR}} = 20$ dB, we conducted experiments with both LARS-LASSO and LARS-EN using randomly selected fixed length exemplars from the training data as described in [17]. For our tuning set and a dictionary size of 2412, 10 iterations of LARS-LASSO gave a recognition rate of 58.70%, and LARS-EN with a sparsity degree of 50 gave a recognition rate of 61.50%. We see that the choice of whole digit exemplars

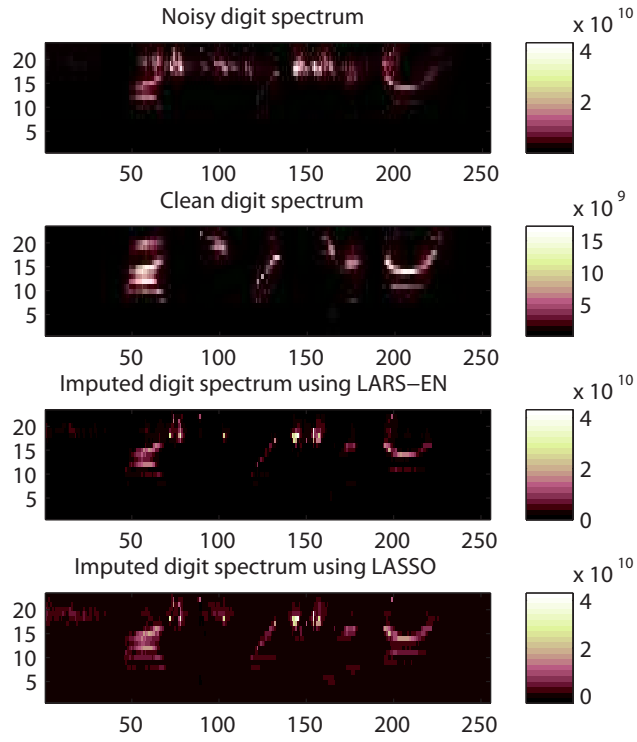

Fig. 5. Spectral plot of one particular imputed utterance using the LARS-LASSO and LARS-EN algorithms for the SNR 5 corruption level. As we can observe from the diagram, the imputed signals have a much closer resemblance to the clean signal than the noisy one with much of the noise artifacts removed. Moreover, we can observe that the imputed result of LARS-EN bears closer resemblance to the clean signal relative to LARS-LASSO, further testifying to the robustness of LARS-EN.

yields a better recognition rate (65.88% for LARS-LASSO and 71.83% for LARS-EN). This can be explained by the fact our dictionary choice retains digit boundary information to a greater extent than the dictionary choice of randomly selected fixed length exemplars. Morever, the transition information is not as significant in the digit recognition setting as compared to the phoneme recognition/LVCSR setting, since the acoustic distance between digits is significantly more distinct as compared to that between phonemes.

2) Investigation of varying dictionary sizes: We next investigate the effectiveness of the various basis selection methods with dictionaries of varying sizes and also investigate the possible relationship to recognition accuracies. For this section, we just consider the results on the dataset with SNR 5 dB corruption. For the dictionary sizes, we consider $N_{\text{train}} = 1000, 1500, 2000$. These dictionaries are random subsets of the original dictionary of size 2412.

From our results in Table IV, it is apparent that the LARS-EN algorithm is consistently better than LARS-LASSO in terms of improving the recognition accuracy of the ASR. Thus LARS-EN does the most robust job in basis selection, regardless of dictionary size. Note that the LARS-EN with a smaller dictionary even outperforms LARS-LASSO with a bigger

TABLE IV
RESULTS FOR SNR 5 dB DATASET WITH $\lambda_{\text{SNR}} = 20$ dB FOR THE AURORA 2.0 DATABASE, WITH $N_{\text{TRAIN}} = 1000, 1500, 2000$ FOR TEST1. THE LAST COLUMN INDICATES WHETHER THE IMPROVEMENT OVER LARS-LASSO IS SIGNIFICANT WITH THE DIFFERENCE OF PROPORTIONS TEST AT 95% CONFIDENCE LEVEL

Alg	Degrees of sparsity	Iterations	Accuracy (%)	Significant?
Unimputed	NA	NA	43.82	NA
$N_{\text{train}} = 1000$				
LARS-LASSO	NA	10	62.62	NA
LARS-EN	50	NA	65.55	YES
$N_{\text{train}} = 1500$				
LARS-LASSO	NA	10	63.47	NA
LARS-EN	50	NA	66.36	YES
$N_{\text{train}} = 2000$				
LARS-LASSO	NA	10	64.01	NA
LARS-EN	50	NA	66.24	YES

dictionary, as evident from Table IV. This demonstration can be useful when we want to port a similar framework to a full LVCSR system rather than the current continuous digits task. This is because optimization in those cases can be expensive in terms of computational power if numerous occurrences of each word have to be included in the dictionary, resulting in an overly large dictionary. Now, with this verification of an efficient basis selection technique for improved recognition, we see that dictionary construction can be a much easier task, since we can choose a smaller subset of the original overcomplete dictionary for our imputation process. Hence, it will definitely be wiser to opt for a smaller dictionary.

IV. DISCUSSION OF PRACTICALITY OF IMPLEMENTATION IN REAL-TIME SYSTEMS

For our dataset of 4004 test files (TEST1), the number of complete optimization problems required to fully impute all windows is anywhere between 10000 and 56577 (upper-limit for our dataset) depending on the value of λ_{SNR} chosen. See Table V for the number of optimization problems for the SNR 5 dB dataset for different values of λ_{SNR}. Specifically, for our threshold of $\lambda_{\text{SNR}} = 20dB$, we have 51260 optimization problems to solve. This is a computationally expensive procedure if we are considering implementation on a real-time system and renders iterative algorithms like the Adaptive LASSO [20] and the Reweighted-L_1 algorithm [9] impractical.

TABLE V
NUMBER OF OPTIMIZATION PROBLEMS FOR THE TEST1 DATASET WITH SNR 5 dB NOISE FOR DIFFERENT VALUES OF λ_{SNR}

λ_{SNR} in dB	Optimizations
0	56568
5	56534
10	56223
15	54536
20	51260
25	44051

The LARS implementation of LASSO and Elastic Net provides an acceleration over classical implementations [18], [33]. For our MATLAB implementation of both LARS-LASSO and

LARS-EN, the entire sparse imputation process for a test utterance generally finishes between 1 to 20 seconds on a Core 2 Quad Processor with 8 gigabytes of RAM for the SNR 5 dB corruption. However, MATLAB is generally slower due to it being a high-level language. When porting this to a real ASR system, we can expect execution time to improve when we are coding in lower level languages such as C. This greatly increases the feasibility of porting our algorithms to an LVCSR system.

V. Conclusion and Extensions

We showed that the LASSO solution for sparse imputation is relatively less effective (theoretically and experimentally) in improving the accuracies of the continuous digit recognition task as compared to the Elastic Net algorithm. The LARS-EN algorithm proved to be the more robust of the two algorithms under our specific experimental conditions (test set, dictionary, training set, parameters choice). We have also seen the effects of appropriate sparsity in helping speech recognition accuracies. The lesson learnt is that it is the quality of the covariates that matters, not the quantity. Moreover, we believe that with appropriate noise models the quality of speech recognition can be improved, and we intend on investigating that in our future work.

An immediate extension to this paper will be to extend our work to a full LVCSR system similar to that described in [18]. The number of spectral segments would increase likewise, and efficient dictionary creation would become a challenge. Thus, basis selection, appropriate noise models, sparsity and algorithmic complexity will play an even more important role in large systems, and the techniques that we proposed to deal with the digits task can be analogously extended to deal with a larger and more general framework. Moreover, for the LVCSR system, it will be useful to explore the effects of dictionary sizes on the overall imputation time and the effects on recognition rates.

For the LVCSR system, transitions between words/phonemes can play a bigger role than in the digits recognition case. Thus, rather than interpolating or doing a random selection of exemplars for dictionary construction, a more informed choice of selecting representative exemplars could potentially lead to improvements of recognition rates. This is a line of work we intend to pursue in future.

Future work on the MFCC sparse imputation front-end can include the investigation of combining dimensionality reduction techniques like HDA [44] with sparse imputation to see if the effects of dimensionality reduction can be capitalized when we are doing basis selection. If we do dimensionality reduction, the number of rows of the dictionary $\mathbf{B}$ will decrease, and thus the number of entries in the basis can be reduced too. With a smaller matrix $\mathbf{B}$, we expect to reduce operation time which will be desirable in a larger system.

Appendix
Evaluation of other regularization/optimization techniques

We implemented the algorithms in Table VI in MATLAB. The SBL algorithm was implemented using the Sparse Bayes toolbox available at http://www.miketipping.com/index.php?page=rvm. The MP and OMP algorithms were implemented using Sparselab available at http://sparselab.stanford.edu.

TABLE VI
Result for SNR 5 dB tuning set with $\lambda_{\text{SNR}} = 20$ dB for the Aurora 2.0 database.

Algorithm	Stopping condition	Accuracy (%)
OMP	N_r iterations max	25.81
SBL	100 iterations max	38.27
MP	100 iterations max	63.30
LASSO	10 iterations	63.64
LARS-EN	Sparsity degree of 50	71.83

Acknowledgment

The authors would like to thank the anonymous reviewers for their time and care taken to review this manuscript, as well as their constructive suggestions.

The research was supported by grants from the NSF, ONR and Army.

References

[1] B. Efron, "Missing Data, Imputation, and the Bootstrap." *Journal of the American Statistical Association*, vol. 89, no. 426, pp. 83–127, 1994.
[2] X. Gan, A. Liew, and H. Yan, "Microarray missing data imputation based on a set theoretic framework and biological knowledge," *Nucleic Acids Research*, vol. 34, no. 5, pp. 1608–1619, 2006.
[3] A. Kokaram and S. Godsill, "A system for reconstruction of missing data in image sequences using sampled 3D AR models and MRF motion priors," *Computer Vision ECCV'96*, pp. 613–624, 1996.
[4] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, pp. 267–285, 2001.
[5] L. Josifovski, M. Cooke, P. Green, and A. Vizinho, "State based imputation of missing data for robust speech recognition and speech enhancement," in *Proc. Eurospeech*, 1999, pp. 2837–2840.
[6] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, p. 1207, 2006.
[7] D. Kanevsky, T. Sainath, B. Ramabhadran, and D. Nahamoo, "An Analysis of Sparseness and Regularization in Exemplar-Based Methods for Speech Classification," 2010.
[8] A. Yang, J. Wright, Y. Ma, and S. Sastry, "Feature selection in face recognition: A sparse representation perspective," *UC Berkeley Technical Report UCB/EECS-2007-99*, 2007.
[9] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l 1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.
[10] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, 2008.
[11] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.
[12] Y. Kim, S. Narayanan, and K. Nayak, "Accelerated three-dimensional upper airway MRI using compressed sensing," *Magnetic Resonance in Medicine*, vol. 61, no. 6, pp. 1434–1440, 2009.
[13] J. F. Gemmeke and B. Cranen, "Using sparse representations for missing data imputation in noise robust speech recognition," in *Proc. of EUSIPCO*, 2008.
[14] J. F. Gemmeke, H. V. Hamme, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 272–286, April 2010.
[15] B. Borgström and A. Alwan, "Missing feature imputation of log-spectral data for noise robust ASR," *Workshop on DSP in Mobile and Vehicular Systems*, 2009.

[16] ——, "Utilizing Compressibility in Reconstructing Spectrographic Data, With Applications to Noise Robust ASR," *IEEE Signal Processing Letters*, vol. 16, no. 5, 2009.
[17] J. Gemmeke and B. Cranen, "Missing data imputation using compressive sensing techniques for connected digit recognition," *Proceedings of DSP 2009*, 2009.
[18] J. Gemmeke, B. Cranen, and U. Remes, "Sparse imputation for large vocabulary noise robust ASR," *Computer Speech & Language*, 2010.
[19] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2005.
[20] H. Zou, "The adaptive LASSO and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.
[21] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.
[22] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[23] H. Hamme, "PROSPECT features and their application to missing data techniques for robust speech recognition," in *Eighth International Conference on Spoken Language Processing*, 2004.
[24] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2008.
[25] G. Lindfield, J. Penny, and J. Penny, *Numerical methods using MATLAB*. Prentice Hall, 1999.
[26] J. Barker, M. Cooke, and P. Green, "Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise," in *Seventh European Conference on Speech Communication and Technology*, 2001.
[27] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech communication*, vol. 34, no. 3, pp. 267–285, 2001.
[28] J. Barker, P. Green, and M. Cooke, "Linking auditory scene analysis and robust ASR by missing data techniques," *Proceedings-Institute of Acoustics*, vol. 23, no. 3, pp. 295–308, 2001.
[29] J. R. Deller, Jr., J. G. Proakis, and J. H. Hansen, *Discrete Time Processing of Speech Signals*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1993.
[30] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006.
[31] E. Candès and M. Wakin, "People hearing without listening: An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.
[33] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.
[34] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, pp. 129–159, 2001.
[35] K. Knight and W. Fu, "Asymptotics for lasso-type estimators," *Annals of Statistics*, vol. 28, no. 5, pp. 1356–1378, 2000.
[36] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *Annals of Statistics*, vol. 37, no. 1, pp. 246–270, 2009.
[37] P. Zhao and B. Yu, "On model selection consistency of LASSO," *The Journal of Machine Learning Research*, vol. 7, p. 2563, 2006.
[38] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
[39] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.
[40] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
[41] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation withapplications to wavelet decomposition," in *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.
[42] T. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 4370–4373.
[43] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," 2000.
[44] N. Kumar and A. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech communication*, vol. 26, no. 4, pp. 283–297, 1998.

Qun Feng Tan is currently pursuing his Ph.D. in Electrical Engineering at the University of Southern California under the Provost Fellowship. He received his B.S.E. in Electrical Engineering with a minor in Mathematics from the University of Michigan, Ann Arbor in 2006, and his M.S. in Electrical Engineering from Stanford University in 2008. His research interest lies in Signal Processing and Pattern Recognition with applications to Speech Processing. Other interests include Recreational Mathematics and Number Theory.

Panayiotis G. Georgiou received his B.A. and M.Eng degrees with Honors from Cambridge University (Pembroke College), U.K. in 1996. He received his MSc and PhD degrees from the University of Southern California in 1998 and 2002 respectively. During the period 1992-96 he was awarded a Commonwealth scholarship from Cambridge-Commonwealth Trust. Since 2003 he has been a member of the Speech Analysis and Interpretation Lab, first as a Research Associate and currently as a Research Assistant Professor. His interests span the fields of Multimodal and Behavioral Signal Processing. He has worked on and published over 80 papers in the fields of statistical signal processing, alpha stable distributions, speech and multimodal signal processing and interfaces, speech translation, language modeling, immersive sound processing, sound source localization, and speaker identification. He has been an Investigator, and co-PI on federally funded projects notably including the DARPA Transtac SpeechLinks and the NSF An Integrated Approach to Creating Enriched Speech Translation Systems. He is currently serving as guest editor of the Computer Speech and Language journal. Papers he has coauthored with his students received best paper awards for analyzing the multimodal behaviors of users in speech- to-speech translation in International Workshop on Multimedia Signal processing (MMSP) 2006 and for automatic classification of married couples behavior using audio features in Interspeech 2010 . His current focus is on multimodal environments, behavioral signal processing, and speech-to-speech translation.

Shrikanth (Shri) Narayanan is the Andrew J. Viterbi Professor of Engineering at the University of Southern California (USC), and holds appointments as Professor of Electrical Engineering, Computer Science, Linguistics and Psychology and as the founding director of the Ming Hsieh Institute. Prior to USC he was with AT&T Bell Labs and AT&T Research from 1995-2000. At USC he directs the Signal Analysis and Interpretation Laboratory (SAIL). His research focuses on human-centered information processing and communication technologies with a special emphasis on behavioral signal processing and informatics. [http://sail.usc.edu]

Shri Narayanan is a Fellow of IEEE, the Acoustical Society of America, and the American Association for the Advancement of Science (AAAS) and a member of Tau-Beta-Pi, Phi Kappa Phi and Eta-Kappa-Nu. Shri Narayanan is also an Editor for the Computer Speech and Language Journal and an Associate Editor for the IEEE Transactions on Multimedia, IEEE Transactions on Affective Computing, and the Journal of the Acoustical Society of America. He was also previously an Associate Editor of the IEEE Transactions of Speech and Audio Processing (2000-04) and the IEEE Signal Processing Magazine (2005-2008). He served on the Speech Processing technical committee (2005-2008) and Multimedia Signal Processing technical committees (2004-2008) of the IEEE Signal Processing Society and presently serves on the Speech Communication committee of the Acoustical Society of America and the Advisory Council of the International Speech Communication Association.

Shri Narayanan is a recipient of a number of honors including Best Paper awards from the IEEE Signal Processing society in 2005 (with Alex Potamianos) and in 2009 (with Chul Min Lee) and selection as an IEEE Signal Processing Society Distinguished Lecturer for 2010-11. He has published over 425 papers and nine granted U.S. patents.