Novel Variations of Group Sparse Regularization Techniques with Applications to Noise Robust Automatic Speech Recognition

Qun Feng Tan* 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,shri@sipi.usc.edu

Abstract

This paper presents novel variations of group sparse regularization techniques. We expand upon the Sparse Group LASSO formulation to incorporate different learning techniques for better sparsity enforcement within a group and demonstrate the effectiveness of the algorithms for spectral denoising with applications to robust Automatic Speech Recognition (ASR). In particular, we show that with a strategic selection of groupings greater robustness to noisy speech recognition can be achieved when compared to state-of-the-art techniques like the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) implementation of the Sparse Group LASSO. Moreover, we demonstrate that group sparse regularization techniques can offer significant gains over efficient techniques like the Elastic Net. We also show that the proposed algorithms are effective in exploiting collinear dictionaries to deal with the inherent highly coherent nature of speech spectral segments. Experiments on the Aurora 2.0 continuous digit database and the Aurora 3.0 realistic noisy database demonstrate the performance improvement with the proposed methods, including showing that their execution time is comparable to FISTA, making our algorithms practical for application to a wide range of regularization problems.

EDICS: SPE-ROBU

Index Terms

Group Sparse Regularization, Sparse Representation, Denoising, Automatic Speech Recognition

I. Introduction

Regularization techniques are commonly employed in statistics, natural sciences, and engineering. In this paper, we are interested in the specific application of regularization techniques to spectral denoising with the aim of improving *Automatic Speech Recognition* [1] (ASR). We begin with a brief survey of relevant efforts in this direction in the field of *Missing Data Techniques* (MDT) in speech. Recently, Gemmeke et al. [2] and Börgstrom et al. [3] have proposed the use of L_1 optimization techniques for spectral denoising in speech recognition. Gemmeke et al. have coined the process "Sparse Imputation" and demonstrated its efficiency over classical missing data techniques. Here, imputation refers to the filling in/substitution/completion of missing data. However, it is well known that L_1 techniques do not generally fair well unless certain properties of the dictionaries are satisfied, and an analysis of these properties is reported in [4]. In particular, we [4] proposed the use of the Elastic Net for better exploiting the characteristics of a coherent dictionary and also provided a rigorous justification of why sparsity is necessary for the improvement of speech recognition rate. We demonstrated significant improvement over LASSO-based strategies for spectral denoising. A block diagram representation of the process is given in Fig. 1.

[Fig. 1 about here.]

In most regularization applications, a linear model is assumed:

$$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{x} \tag{1}$$

Here, $\boldsymbol{f}$ is the observed feature vector, $\boldsymbol{\Phi}$ is a basis/dictionary, and $\boldsymbol{x}$ is the activation. Frequently, the goal is to find a sparse or maximally sparse $\boldsymbol{x}$ which best reconstructs $\boldsymbol{f}$. By sparse signals, we mean signals that are zero everywhere except on a minimal support of the solution space [5]. Ideally, we would like to solve the following optimization problem which optimizes the L_p norm (the L_p norm is defined as $||\boldsymbol{x}||_p = (\sum_{k=1}^{n} |x_k|^p)^{\frac{1}{p}}$):

$$\min_{\boldsymbol{x}} \|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{f}\|_2^2 + \lambda \|\boldsymbol{x}\|_p \tag{2}$$

For a maximally sparse solution vector, we ideally would like to solve the following:

$$\min_{\boldsymbol{x}} \|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{f}\|_2^2 + \lambda \|\boldsymbol{x}\|_0 \tag{3}$$

However, it is well known that this problem is NP-hard, since solving Equation (3) will involve searching through $\binom{n}{k}$ least-square problems, where n denotes the dimensions of the activation x, assuming that x is k-sparse (having k nonzero activations). However, there are some good greedy-based solutions which seek to approximate the solution to Equation (3), with examples including *Matching Pursuit* (MP) [6] and *Orthogonal Matching Pursuit* (OMP) [7].

It is typical that we consider a convex relaxation of the above problem to optimize the L_1 norm instead:

$$\min_{\boldsymbol{x}} \|\boldsymbol{\Phi x} - \boldsymbol{f}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{4}$$

There exist many efficient solutions for solving Equation (4); examples include the *Least Absolute Shrinkage and Selection Operator* (LASSO) [8] and the *Least Angle Regression* (LARS) algorithm [9].

In this paper, we are more interested in a particular extension of the LASSO model to incorporate grouping information, given by the following:

$$\min_{\boldsymbol{x}} \left\| \sum_{i=1}^{N} \boldsymbol{\Phi}_i \boldsymbol{x}_i - \boldsymbol{f} \right\|_2^2 + \lambda \sum_{l=1}^{N} \|\boldsymbol{x}_l\|_1 \tag{5}$$

Here, $\boldsymbol{\Phi}_i$ stands for partitions of the dictionary and $\boldsymbol{x}_i$ stands for the corresponding activations. Yuan and Lin [10] proposed an efficient solution for resolving Equation (5). Meier et al. [11] have proposed a variation which solves the problem efficiently for logistic regression models. However, it has been pointed out in [12] that the group LASSO does not yield sparsity within a group. Thus, whenever a group has some non-zero parameters, they will likely be all non-zero. Hence, the proposal is to consider the following problem formulation instead:

$$\min_{\boldsymbol{x}} \left\| \sum_{i=1}^{N} \boldsymbol{\Phi}_i \boldsymbol{x}_i - \boldsymbol{f} \right\|_2^2 + \lambda_1 \sum_{l=1}^{N} \|\boldsymbol{x}_l\|_1 + \lambda_2 \sum_{l=1}^{N} \|\boldsymbol{x}_l\|_2 \tag{6}$$

This is known as the *Sparse Group LASSO* (SGL). It is a more general formulation than the Group LASSO since, when $\lambda_2 = 0$, we will have the Group LASSO. Experimental simulation in [12] has provided promising evidence of a more robust performance in the presence of collinear dictionaries over the Group LASSO and the LASSO algorithms. While [12] has proposed a one-dimensional optimization based on the Golden Section Search algorithm to capitalize on the separable penalty function, there are other efficient solutions to the formulation in Equation (6). Notably, the software SLEP [13] implements the SGL formulation using a version of *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA) [14].

The first contribution of this paper is to propose novel variations along the lines of the SGL formulation to better enforce sparsity within a group. Specifically, we propose two algorithms, one which follows the theme of the Least Angle Regression implementation of the Elastic Net [15] and the other along the lines of *Sparse Bayesian Learning* (SBL) [16]. The SBL algorithm boasts a prior which is able to enforce sparsity efficiently and in this paper we will experimentally justify this for our spectral denoising task for improving speech recognition.

The second contribution of this paper is the study of the effects of grouping spectral atoms in the dictionary on speech recognition rates. We discuss grouping techniques for enhancing the spectral denoising process for a dictionary consisting of a variety of spectral exemplars. In particular, we explore groupings based on speaker identity and also based on L_2 distance between the feature vectors. Through experiments on the Aurora 2.0 noisy digits database and the Aurora 3.0 real noisy data, we demonstrate that clustering based on either strategy will lead to an appreciable increase in speech recognition rates. In fact, experimental evidence show that these grouping strategies, coupled with an appropriate group sparse regularization technique, yield better accuracies than the Elastic Net algorithm.

The structure of the paper is as follows: Section II and III details the derivation of our proposed algorithms, theoretical justification as to the suitability of these algorithms in the denoising framework, and a description of the feature extraction process. Section IV and V describes our dataset, experimental results and interpretation of the results. Finally, Section VI concludes with some possible consequences and extensions of our work in this paper.

II. Methodology

A. Notation description

Before delving into the main results of the paper, we first introduce the notational conventions adopted in the paper.

For matrices, we use a bold uppercase font - for example, $\mathbf{F}$ denotes a matrix of features; in speech recognition the columns typically would correspond to sequence of spectral frame information (See Sec IV). For vectors, we use a bold lowercase font - for example $\boldsymbol{f}$ refers to a vector of features. Otherwise, a normal font refers to a scalar value - for example, f can refer to a particular entry in our feature vector.

When we add a hat on top of a variable, unless stated otherwise, we refer to a result returned from an optimization formulation, for example, $\widehat{\boldsymbol{x}}$ can refer to the solution returned from Equation (4).

B. Basis Selection Techniques

1) Partitioning the dictionary: In order to form our dictionary $\mathbf{\Phi} = [\phi_1\ \phi_2\ ...\ \phi_{N_{\text{train}}}]$, where N_{train} is the number of training samples in the dictionary, we stack spectral exemplars as the columns of the dictionary.

In order to further capitalize on the structure of the dictionary, we assume that we can partition the dictionary as

$$\begin{aligned} \mathbf{\Phi} &= [\phi_1\ \phi_2\ ...\ \phi_{N_{\text{train}}}] \\ &= [\mathbf{\Phi}_1\ |\mathbf{\Phi}_2|\ ...\ |\mathbf{\Phi}_{N_C}] \end{aligned} \tag{7}$$

In equation (7), "|" represents the partition boundaries, $\mathbf{\Phi}_i$ is a matrix comprised of some columns from $\mathbf{\Phi}$, and N_C denotes the number of partitions (clusters) we have. When $N_C = N_{\text{train}}$, we just have each partition consist of a column of the dictionary. Later sections in this paper will describe efficient partitioning schemes that will be crucial for improving speech recognition rates.

In view of the dictionary partitioning, we can reformulate our problem in Equation (2) as the following:

$$\min_{\boldsymbol{x}} \left\| \sum_{l=1}^{N_C} \mathbf{\Phi}_l \boldsymbol{x}_l - \boldsymbol{f} \right\|_2^2 + \lambda \sum_{l=1}^{N_C} s_l \left\| \boldsymbol{x}_l \right\|_p \tag{8}$$

s_l refers to a penalty rescaling factor with respect to the dimensionality of $\boldsymbol{x}_l$.

2) Better exploitation of a collinear dictionary: In most speech data, we can expect the spectral profiles for the same phonemes/words (in general, a sound unit) to be similar. Thus we can expect groups of contiguous entries in the dictionary to be highly collinear. As a result of the collinear dictionary, a method like LASSO will not select the relevant atoms efficiently due to the fact that it does not discriminate between the collinear entries well enough [4]. In particular, let us define the covariance matrix $\mathbf{C}$ to be:

$$\mathbf{C} = \frac{1}{\dim(\mathbf{\Phi})} \mathbf{\Phi}^T \mathbf{\Phi} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix} \tag{9}$$

where $\mathbf{C}$ is a positive definite matrix. Suppose we arrange the columns of $\mathbf{\Phi}$ such that $\mathbf{\Phi} = \left[\phi_1 ... \phi_{N_{train}}\right]$ and $\phi_1, ..., \phi_k$ corresponding to the k non-zero activations. We set

$$\begin{aligned}
\mathbf{C}_{11} &= \frac{1}{\dim(\mathbf{\Phi})} [\boldsymbol{\phi}_1 \ ... \ \boldsymbol{\phi}_k]^T [\boldsymbol{\phi}_1 \ ... \ \boldsymbol{\phi}_k] \\
\mathbf{C}_{22} &= \frac{1}{\dim(\mathbf{\Phi})} \left[\boldsymbol{\phi}_{k+1} \ ... \ \boldsymbol{\phi}_{N_{\text{train}}}\right]^T \left[\boldsymbol{\phi}_{k+1} \ ... \ \boldsymbol{\phi}_{N_{\text{train}}}\right] \\
\mathbf{C}_{12} &= \frac{1}{\dim(\mathbf{\Phi})} [\boldsymbol{\phi}_1 \ ... \ \boldsymbol{\phi}_k]^T \left[\boldsymbol{\phi}_{k+1} \ ... \ \boldsymbol{\phi}_{N_{\text{train}}}\right] \\
\mathbf{C}_{21} &= \frac{1}{\dim(\mathbf{\Phi})} \left[\boldsymbol{\phi}_{k+1} \ ... \ \boldsymbol{\phi}_{N_{\text{train}}}\right]^T [\boldsymbol{\phi}_1 \ ... \ \boldsymbol{\phi}_k] \qquad (10)
\end{aligned}$$

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

$$P(\text{sign}(\boldsymbol{x}) = \text{sign}(\widehat{\boldsymbol{x}})) \rightarrow 1 \text{ as } \text{rank}(\mathbf{\Phi}) \rightarrow \infty \qquad (11)$$

Sign consistency is a necessary condition for the LASSO estimate to match the true model. It has been demonstrated in [17], [18] that LASSO is sign consistent only if $|\mathbf{C}_{21}\mathbf{C}_{11}^{-1}\text{sign}([\widehat{\boldsymbol{x}}_1 \ ... \ \widehat{\boldsymbol{x}}_p]^T)| \leq \mathbf{1} - \eta$ (Strong Irrepresentable Condition). However, this condition has a high possibility of being violated when the columns of $\mathbf{\Phi}$ are highly collinear, a pressing issue in contending with speech data.

The work in [4] has shown that the Elastic Net significantly outperforms LASSO in terms of speech recognition accuracies in the spectral denoising framework by being able to better exploit the properties of a collinear dictionary. Hence, when we are considering group sparse regularization techniques, this will be motivation to consider algorithms along the lines of the SGL algorithm [12] rather than the Group LASSO algorithm [10]. Moreover, as the authors in [12] pointed out, the Group LASSO algorithm does not yield sparsity within a group. In particular, when a group of activations is dropped the entire group is dropped, and when a group is non-zero they will all be non-zero.

C. Formulation of the "Group Elastic Net" algorithm

Following the background discussed in Section II-B2, we now set out to formulate the "Group Elastic Net" and the "Group Sparse Bayesian Learning" algorithms for our problem setup. In particular, the first formulation extends the Elastic Net formulation to a more general setting, while retaining the speed (complexity) advantages of the Elastic Net, and the second formulation further promotes sparsity enforcement within a partition.

Motivated by Equation (7) and the Elastic Net, we can have a different formulation of our optimization problem as follows:

$$\min_{\boldsymbol{x}} \left\| \sum_{l=1}^{N_C} \boldsymbol{\Phi}_l \boldsymbol{x}_l - \boldsymbol{f} \right\|_2^2 + \lambda_2 \sum_{l=1}^{N_C} s_l \left\| \boldsymbol{x}_l \right\|_2^2 + \lambda_1 \sum_{l=1}^{N_C} r_l \left\| \boldsymbol{x}_l \right\|_1 \tag{12}$$

r_l and s_l are rescaling factors for the L_1 norm and L_2 norm, respectively. Due to the fact that our penalty function is separable, we can take the usual route of considering each group as an individual optimization problem. Suppose at the current iteration we are considering group i. Define the following:

$$\begin{aligned} \boldsymbol{\Phi}_i &= \begin{pmatrix} \boldsymbol{\phi}_{i,1} & \boldsymbol{\phi}_{i,2} & ... & \boldsymbol{\phi}_{i,N_i} \end{pmatrix} \\ \boldsymbol{x}_i &= \begin{pmatrix} x_{i,1} & x_{i,2} & ... & x_{i,N_i} \end{pmatrix}^T \end{aligned} \tag{13}$$

N_i denotes the number of columns in group i. Let us define the residual $\boldsymbol{r} = \boldsymbol{f} - \sum_{j \neq i} \boldsymbol{\Phi}_j \boldsymbol{x}_j$.

We will then need to solve a successive series of optimization problems of the following form:

$$\min ||\boldsymbol{r} - \boldsymbol{\Phi}_i \boldsymbol{x}_i||_2^2 + \lambda_2 s_i ||\boldsymbol{x}_i||_2^2 + \lambda_1 r_i ||\boldsymbol{x}_i||_1 \tag{14}$$

Define $\boldsymbol{r}'$ to be a $(\dim(\boldsymbol{r}) + \dim(\boldsymbol{x}_i)) \times 1$ vector as follows:

$$\boldsymbol{r}' = \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{0} \end{pmatrix} \tag{15}$$

We can manipulate the objective function (denote it J) in Equation (14) as follows:

$$\begin{aligned} J &= ||\boldsymbol{r} - \boldsymbol{\Phi}_i \boldsymbol{x}_i||_2^2 + \lambda_2 s_i ||\boldsymbol{x}_i||_2^2 + \lambda_1 r_i ||\boldsymbol{x}_i||_1 \\ &= ||\boldsymbol{r}' - \begin{pmatrix} \boldsymbol{\Phi}_i \\ \sqrt{\lambda_2 s_i}\mathbf{I} \end{pmatrix} \boldsymbol{x'}_i||_2^2 + \lambda_1 r_i ||\boldsymbol{x'}_i||_1 \\ &= ||\boldsymbol{r}' - \frac{1}{\sqrt{1+\lambda_2 s_i}} \begin{pmatrix} \boldsymbol{\Phi}_i \\ \sqrt{\lambda_2 s_i}\mathbf{I} \end{pmatrix} \boldsymbol{x''}_i||_2^2 + \\ &\quad \frac{\lambda_1 r_i}{\sqrt{1+\lambda_2 s_i}} ||\boldsymbol{x''}_i||_1 \end{aligned} \tag{16}$$

The solution to Equation (16) gives the solution to the original optimization problem stated in Equation (14). Specifically, we can scale $\widehat{\boldsymbol{x}''_i}$ to give $\widehat{\boldsymbol{x}'_i} = \frac{1}{\sqrt{1+\lambda_2 s_i}} \widehat{\boldsymbol{x}''_i}$ and $\widehat{\boldsymbol{x}}_i$ is equal to $\widehat{\boldsymbol{x}'_i}$. Moreover, the formulation in Equation (16) increases the rank of the dictionary (and hence reduces the coherence), which will contribute to mitigating the issue of highly coherent dictionary atoms.

If we proceed to solve Equation (16) by the LASSO algorithm, we will have the Elastic Net formulation [15]. This can be further sped up to be equivalent to the order of one *Ordinary Least Squares* (OLS) fit by employing a modified version of the LARS-LASSO solution [9]. We will call this method the "Group Elastic Net" algorithm (Group EN) and the gist of the algorithm is given in Algorithm 1.

Algorithm 1 "Group Elastic Net" Algorithm

1: Initialize the algorithm with $\boldsymbol{x} = \boldsymbol{\Phi}^{+}\boldsymbol{f}$. Here, we are intializing with the least squares solution and $\boldsymbol{\Phi}^{+}$ is the Pseudo-Inverse.

2: For each group $i = 1...N_C$, we define the residual $\boldsymbol{r} = \boldsymbol{f} - \sum_{j \neq i} \boldsymbol{\Phi}_j \boldsymbol{x}_j$. We proceed to the Elastic Net algorithm to solve the following optimization problem:

$$\min_{x_i} ||\boldsymbol{r} - \boldsymbol{\Phi}_i \boldsymbol{x}_i||_2^2 + \lambda_1 s_i ||\boldsymbol{x}_i||_2^2 + \lambda_2 r_i ||\boldsymbol{x}_i||_1 \tag{17}$$

3: Iterate over all the groups repeatedly until convergence is reached.

D. Formulation of "Group Sparse Bayesian Learning" algorithm

Since our goal in the grouped regularization setting is to enforce sparsity within a group as well as between the groups, there are alternatives to LARS-LASSO for resolving the optimization of Equation (16). In particular, here we advocate the use of *Sparse Bayesian Learning* [16], [19] due to the fact that it makes use of a data parametrized prior which can be effective for enforcing sparsity. By assuming a parametrized prior, a good measure of sparsity relative to both L_1 and L_2 optimization can be obtained.

We assume a Gaussian likelihood model for the residual:

$$p(\boldsymbol{r}'|\boldsymbol{x}_i'', \sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{d}{2}} exp\left(-\frac{1}{2\pi\sigma^2}\left|\left|\frac{1}{\sqrt{1+\lambda_2 s_i}}\begin{pmatrix}\boldsymbol{\Phi}_i \\ \sqrt{\lambda_2 s_i}\mathbf{I}\end{pmatrix}\boldsymbol{x}_i'' - \boldsymbol{r}'\right|\right|^2\right) \tag{18}$$

where $d = \dim(\boldsymbol{r}) + \dim(\boldsymbol{x}_i)$. The SBL also assumes a parametrized prior from the training data, which is given by

$$p(\boldsymbol{x}_i''|\boldsymbol{\gamma}) = \prod_{j=1}^{N_i} \frac{1}{(2\pi\gamma_j)^{\frac{1}{2}}} e^{-\frac{\boldsymbol{x}_{i_j}''^2}{2\gamma_j}} \tag{19}$$

Here, $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2\ ...\ \gamma_{N_i}]$ are the hyperparameters which regulate the prior variance of each weight. The inverse of the hyperparameters are chosen to be distributed according to a gamma distribution [16]:

$$p(\boldsymbol{\gamma}^{-1}) = \prod_{j=1}^{N_i} \Gamma(\gamma_j^{-1}|m_\gamma, n_\gamma) \tag{20}$$

m_γ and n_γ are the parameters of the Gamma distribution. We will see later (in Property 1) that $\boldsymbol{\gamma}$ controls the degree of sparsity of the vector $\boldsymbol{x}_i$, and thus can be likened to the role of λ_1. However, since $\boldsymbol{\gamma}$ is adapted at each iteration of the algorithm described below, we do not have to concern ourselves with the explicit form of $\boldsymbol{\gamma}$.

By choosing the appropriate prior as in (19), the posterior density of the activation weights $\boldsymbol{x}_i$ will be distributed according to a Gaussian, which is given by:

$$p(\boldsymbol{x}_i''|\boldsymbol{r}, \boldsymbol{\gamma}, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\boldsymbol{x}_i''}) \tag{21}$$

with

$$\boldsymbol{\mu} = \frac{1}{\sigma^2\sqrt{1+\lambda_2 s_i}} \boldsymbol{\Sigma}_{\boldsymbol{x}_i''} \left[\boldsymbol{\Phi}_i^T \ \sqrt{\lambda_2 s_i}\mathbf{I}\right] \boldsymbol{r} \tag{22}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{x_i}''} = \left[\frac{1}{\sigma^2(1+\lambda_2 s_i)}(\boldsymbol{\Phi}_i^T\boldsymbol{\Phi}_i + \lambda_2 s_i \mathbf{I}) + \text{diag}(\boldsymbol{\gamma}^{-1})\right]^{-1} \tag{23}$$

To compute the cost function for SBL, note that to find $p(\boldsymbol{r'}|\boldsymbol{\gamma}, \sigma^2)$, we marginalize to get:

$$\begin{aligned} p(\boldsymbol{r'}|\boldsymbol{\gamma}, \sigma^2) &= \int p(\boldsymbol{r'}|\boldsymbol{x}_i, \sigma^2) p(\boldsymbol{x}_i|\boldsymbol{\gamma}) d\boldsymbol{x}_i \\ &= \frac{1}{2\pi^{\frac{d}{2}}|\Sigma_{\boldsymbol{r'}}|^{\frac{1}{2}}} e^{-\frac{1}{2}\boldsymbol{r'}^T\Sigma_{\boldsymbol{r'}}^{-1}\boldsymbol{r'}} \end{aligned} \tag{24}$$

Here, we have

$$\begin{aligned} \Sigma_{\boldsymbol{r'}} &= \sigma^2 I + \sum_{j=1}^{N_{train}} \frac{\gamma_j}{1+\lambda_2 s_i} \begin{pmatrix} \boldsymbol{\Phi}_{i,j}\boldsymbol{\Phi}_{i,j}^T & \sqrt{\lambda_2 s_i}\boldsymbol{\Phi}_{i,j}\mathbf{I}_j \\ \sqrt{\lambda_2 s_i}\mathbf{I}_j\boldsymbol{\Phi}_{i,j}^T & \lambda_2 s_i \mathbf{I}_j \end{pmatrix} \\ &= \sigma^2 I + \sum_{j=1}^{N_{train}} \frac{\gamma_j}{1+\lambda_2 s_i} \mathbf{B}_j \end{aligned} \tag{25}$$

$\mathbf{I}_j$ denotes an indicator matrix of all zeros except at the (j, j) location, where the value is 1. To maximize $p(\boldsymbol{r'}|\boldsymbol{\gamma}, \sigma^2)$, we will be equivalently minimizing $-\log p(\boldsymbol{r'}|\boldsymbol{\gamma}, \sigma^2)$. Thus, our cost function is:

$$-\log p(\boldsymbol{r'}|\boldsymbol{\gamma},\sigma^2) \propto \log|\Sigma_{\boldsymbol{r'}}| + \boldsymbol{r'}^T\Sigma_{\boldsymbol{r'}}^{-1}\boldsymbol{r'} \tag{26}$$

Minimizing the expression in Equation (26) will give us an update rule of the hyperparameters. Alternatively, we can treat the activations $\boldsymbol{x}_i$ as hidden variables [16], [19] and then maximize $\mathrm{E}_{\boldsymbol{x}_i''|\boldsymbol{r'},\boldsymbol{\gamma},\sigma^2}\left[p(\boldsymbol{r'},\boldsymbol{x}_i'',\boldsymbol{\gamma},\sigma^2)\right]$ to give an *Expectation Maximization* (EM) formulation for SBL.

However, it has been demonstrated by Tipping et al. in [20] that this EM implementation of the SBL algorithm is generally slower than if we considered the atoms in the dictionary sequentially. In fact, experimental verification in [20] has demonstrated that *the Sequential Sparse Bayesian Learning* (SSBL) performs more than 15 times faster as compared to the EM implementation. In particular, for our case, this speedup will be even more relevant for two reasons: firstly, our dictionary is augmented to be even larger as given in Equation (16); secondly, our algorithm consists of a nested loop, which requires a fast algorithm to be practically feasible.

Following the trend in [20], [21], we are able to write Equation (25) as:

$$\begin{aligned}\Sigma_{\boldsymbol{r'}} &= \sigma^2 I + \sum_{j=1,j\neq k}^{N_{train}} \frac{\gamma_j}{1+\lambda_2 s_i}\mathbf{B}_j + \frac{\gamma_k}{1+\lambda_2 s_i}\mathbf{B}_k \\ &= \Sigma'_{\boldsymbol{r'}} + \frac{\gamma_k}{1+\lambda_2 s_i}\mathbf{B}_k \end{aligned} \tag{27}$$

For subsequent expression simplification, let us define the following variables:

$$\begin{aligned} c_k &= r'^T\Sigma'^{-1}_{\boldsymbol{r'}}\mathbf{B}_k\Sigma'^{-1}_{\boldsymbol{r'}}r' \\ d_k &= \left[\mathbf{\Phi}_k^T \ \sqrt{\lambda_2 s_i}\mathbf{I}\right]{\Sigma'_{\boldsymbol{r'}}}^{-1}\left[\mathbf{\Phi}_k \ \sqrt{\lambda_2 s_i}\mathbf{I}\right] \end{aligned} \tag{28}$$

After algebraic manipulations, we can write the right-hand side of Equation (26) as:

$$\begin{aligned} P &= \log|\Sigma_{\boldsymbol{r'}}| + \boldsymbol{r'}^T\Sigma_{\boldsymbol{r'}}^{-1}\boldsymbol{r'} \\ &= \log|\gamma_k\Sigma'_{\boldsymbol{r'}}(\frac{1}{\gamma_k} + \frac{1}{1+\lambda_2 s_i}\Sigma'^{-1}_{\boldsymbol{r'}}\mathbf{B}_k)| + \boldsymbol{r'}^T\Sigma_{\boldsymbol{r'}}^{-1}\boldsymbol{r'} \\ &= \log|\Sigma'_{\boldsymbol{r'}}| + \boldsymbol{r'}^T\Sigma'^{-1}_{\boldsymbol{r'}}\boldsymbol{r'} + \log\gamma_k \\ &\quad + \log\left(\frac{1}{\gamma_k} + \frac{1}{1+\lambda_2 s_i}d_k\right) \\ &\quad - \frac{c_k}{\frac{1+\lambda_2 s_i}{\gamma_k} + d_k} \end{aligned} \tag{29}$$

Note that inverse of $\Sigma_{\boldsymbol{r}'}^{-1}$ can be found using the Woodbury Identity. Taking partial derivatives w.r.t γ_k^{-1},

$$\begin{aligned}\frac{\partial P}{\partial \gamma_k^{-1}} =& -\frac{1}{\gamma_k^{-1}} + \frac{1+\lambda_2 s_i}{\gamma_k^{-1}(1+\lambda_2 s_i)+d_k} \\ &+ \frac{c_k(1+\lambda_2 s_i)}{\left[\gamma_k^{-1}(1+\lambda_2 s_i)+d_k\right]^2}\end{aligned} \tag{30}$$

Setting $\frac{\partial P}{\partial \gamma_k^{-1}}$ to be zero and denoting $\Delta = \gamma_k^{-1}(1+\lambda_2 s_i)$, we have

$$\begin{aligned}&-\frac{1}{\gamma_k^{-1}} + \frac{1+\lambda_2 s_i}{\gamma_k^{-1}(1+\lambda_2 s_i)+d_k} + \frac{c_k(1+\lambda_2 s_i)}{\left[\gamma_k^{-1}(1+\lambda_2 s_i)+d_k\right]^2} = 0 \\ &-\frac{1}{\Delta} + \frac{1}{\Delta+d_k} + \frac{c_k}{(\Delta+d_k)^2} = 0 \\ &-(\Delta+d_k)^2 + \Delta^2 + d_k\Delta + c_k\Delta = 0 \\ &-d_k^2 - \Delta d_k + c_k\Delta = 0 \\ &\Delta = \frac{d_k^2}{c_k - d_k} \\ &\gamma_k^{-1}(1+\lambda_2 s_i) = \frac{d_k^2}{c_k - d_k} \\ &\gamma_k^{-1} = \frac{d_k^2}{(1+\lambda_2 s_i)(c_k - d_k)}\end{aligned} \tag{31}$$

We see that when $c_k - d_k < 0$, γ_k^{-1} is positive. However, when $c_k - d_k \leq 0$, γ_k^{-1} is undefined; thus we default it to infinity.

Thus, from the analysis of the stationary points, we can see that Equation (29) has a minimum w.r.t γ_k given by:

$$\gamma_k^{-1} = \begin{cases} \infty & \text{if } c_k \leq d_k \\ \frac{d_k^2}{(1+\lambda_2 s_i)(c_k - d_k)} & \text{otherwise} \end{cases} \tag{32}$$

In fact, when the parameters of the Gamma distribution in Equation (20) m_γ, n_γ approach zero, we will have $p(\boldsymbol{x}_i'') \rightarrow c\prod_{j=1}^{N_{train}} \frac{1}{|\boldsymbol{x}_{i\,j}''|}$ where c is some constant [16]. We now state a property that justifies why we selected the SBL algorithm to enforce sparsity within a group:

Property 1. (SBL Sparsity Property) *When the parameters of the Gamma distribution in Equation (20) m_γ, n_γ approach zero, we have $p(\boldsymbol{x}_i'') \rightarrow c\prod_{j=1}^{N_i} \frac{1}{|\boldsymbol{x}_{ij}''|}$ where c is some constant. (See Appendix for derivation)*

This evidently induces greater sparsity since the distribution peaks sharply at zero and has heavy tails. Moreover, from the analysis given in [19], we know that even when the algorithm gets to a local minima instead of the desired global minima, we will still obtain a solution with maximal sparsity.

Thus, we can now formulate the "Group Sparse Bayesian Learning" (Group SBL) algorithm as summarized in Algorithm 2.

Algorithm 2 "Group Sparse Bayesian Learning" (Group SBL) Algorithm

1: Initialize the algorithm with $\boldsymbol{x} = \boldsymbol{\Phi}^{+}\boldsymbol{f}$. Here, we are intializing with the least squares solution and $\boldsymbol{\Phi}^{+}$ is the Pseudo-Inverse.

2: For each group $i = 1...N_C$, we define the residual $\boldsymbol{r} = \boldsymbol{f} - \sum_{j \neq i} \boldsymbol{\Phi}_j \boldsymbol{x}_j$. We then proceed as follows:

3: Initialize σ^2

4: Initialize for some $j \in \{1...N_{train}\}$, $\gamma_j^{-1} = \frac{||\boldsymbol{\Phi}_{i,j}||^2 + \lambda_2 s_i}{(1+\lambda_2 s_i)\left(\frac{||[\boldsymbol{\Phi}_{i,j}^T \quad \sqrt{\lambda_2 s_i}\mathbf{I}_j]\boldsymbol{r}||^2}{||\boldsymbol{\Phi}_{i,j}||^2 + \lambda_2 s_i} - \sigma^2\right)}$, and $\gamma_k^{-1} = \infty$ if $k \neq j$

5: Compute $\Sigma_{\boldsymbol{x}_i''}$ and $\boldsymbol{\mu}$.

6: For $j = 1...N_{train}$, we check the following: If $c_j > d_j$ and $\gamma_j^{-1} < \infty$, we re-estimate γ_i^{-1}. Else if $c_j > d_j$ and $\gamma_j^{-1} = \infty$, we add $\boldsymbol{\Phi}_{i,j}$ to the model. Otherwise, if $c_j \leq d_j$ and γ_j^{-1} is finite, remove $\boldsymbol{\Phi}_{i,j}$ and set γ_j^{-1} to infinity. Re-estimate $\Sigma_{\boldsymbol{x}_i''}$ and $\boldsymbol{\mu}$ from the relevant updated model. Repeat until convergence or some iteration limit.

7: Iterate over all the groups until convergence is reached.

III. Application to Spectral Denoising in a Speech Recognition Framework

Now that we have described the derivations and details of the algorithms, we will consider an application of the group sparse regularization techniques in the setting of spectral denoising within a speech recognition framework [1]. In particular, for ASR, we need to extract features from the audio data, which constitute the speech recognition front end. Fig. 1 shows a typical speech recognition system, but with a modified front-end which incorporates a spectral denoiser. The spectral denoiser will incorporate the group sparse regularization techniques with appropriate dictionary partitioning. However, the original feature matrices that we extract from the audio will have a variable number of columns (dependent on the duration of the audio), and thus we will need an alternative feature extraction procedure which we will now describe.

We consider a framework for extraction of features as in [4]. Denote the number of frames for an utterance as T. Let the feature vector corresponding to frame k in an utterance be represented by $\boldsymbol{f}_k$,

where $\boldsymbol{f}_k$ is a $N_B \times 1$ dimensional vector which contains the spectral coefficients, and N_B is the number of frequency bands. Define $\tilde{\mathbf{F}}$ to be a $N_B \times T$ matrix as follows:

$$\tilde{\mathbf{F}} = \begin{pmatrix} \boldsymbol{f}_1 & \boldsymbol{f}_2 & \cdots & \boldsymbol{f}_T \end{pmatrix} \tag{33}$$

We define the linearization of a matrix $\mathbf{F} = [\boldsymbol{f}_1 ... \boldsymbol{f}_n]$ as follows:

$$\bar{\boldsymbol{f}} = \begin{pmatrix} \boldsymbol{f}_1 \boldsymbol{f}_2 \cdots \boldsymbol{f}_n \end{pmatrix}^T \tag{34}$$

When we try to linearize $\tilde{\mathbf{F}}$, we will obtain vectors of different lengths due to the different audio durations. Thus, we need to consider the features in blocks to ensure conformity of dimensions. Consider a sliding window extraction of the data in this matrix representation $\tilde{\mathbf{F}}$. Define a sliding matrix (window) which has dimensions $N_B \times T_W$, T_W represents the length in frames of the sliding matrix. We also define a shift parameter T_S, which represents the number of frames by which we shift the sliding matrix.

In doing so, we obtain a total of $\lceil \frac{(T-T_W)}{T_S} \rceil + 1$ matrices of feature vectors. We zero-pad $\tilde{\mathbf{F}}$ to be a $N_B \times c$ matrix where $c = \lceil \frac{(T-T_W)}{T_S} \rceil \times T_S + T_S$. Let us denote the m-th sliding window to be $\tilde{\mathbf{F}}_m = \begin{pmatrix} \boldsymbol{f}_{1+(m-1)T_S} & \cdots & \boldsymbol{f}_{T_W+(m-1)T_S} \end{pmatrix}$.

We now suppose that we can write $\bar{\boldsymbol{f}}_m$ as $\bar{\boldsymbol{f}}_m = \boldsymbol{\Phi} \boldsymbol{x}_m$, where $\bar{\boldsymbol{f}}_m$ is the observation (feature vector) linearized from $\tilde{\mathbf{F}}_m$, $\boldsymbol{\Phi}$ is the dictionary of exemplars, and $\boldsymbol{x}_m$ is a vector of activation. We are assuming that each test segment can be written as a linear combination of the basis vectors. This is a reasonable assumption since the spectral representations for different realizations of the same word have energy localizations in similar regions in the time-frequency domain.

We thus obtain the following type of linear representations from our windows:

$$\bar{\boldsymbol{f}}_m = \boldsymbol{\Phi} \boldsymbol{x}_m, \quad m = 1, ..., \left\lceil \frac{T - T_W}{T_S} \right\rceil + 1 \tag{35}$$

After the sparse imputation process, we need to reconstruct a denoised representation of the original sliding matrix. Define a counter matrix of dimension $N_B \times c$ where c is defined as above. This counter matrix counts the number of times each entry in the matrix $\tilde{\mathbf{F}}_m$ is optimized due to overlapping window shifts. Formation of the final denoised matrix will involve first reshaping $\widehat{\tilde{\mathbf{F}}_m}$ (the solution to optimizing Equation (35)) back to dimensions $N_B \times c$, summing all the resulting reshaped frames, and then doing component-wise division by the entries of the counter matrix.

Let us denote the number of columns in our dictionary by N_{train}. This is the number of linearized exemplars used for the spectral denoising process. We then form a dictionary $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \ ... \ \boldsymbol{\phi}_{N_{\text{train}}}]$

which consists of segments of clean spectral shapes. This will be our overcomplete dictionary of exemplar spectral segments. Section IV will discuss procedures by which we obtain $\mathbf{\Phi}_i$ for $i = 1, ..., N_{\text{train}}$ as efficiently as possible.

IV. Experimental Setup

A. Description of Database

For the Aurora 2.0 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 [22].

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 merge all audio files from each of N1, N2, N3, and N4 to give us a total of 4004 files in our test set for a specific noise SNR. We will be using SNR levels -5 dB, 0 dB, 5 dB, and 10 dB.

To form our dictionary $\mathbf{\Phi}$, we get all the single digit audio files in the training data, extract the spectral features corresponding to those files, and then construct a dictionary of fixed window-size spectral exemplars (window length T_W) comprising whole digit interpolated spectral exemplars (magnitude domain) like in [4]. We then linearize these spectral exemplars which will go into the columns of our dictionary $\mathbf{\Phi}$.

Since the Aurora 2.0 is a synthetically corrupted dataset, in order to further stress test our algorithms, we evaluated them on the Aurora 3.0 database as well. We trained an Italian continuous digit recognizer using 961 continuous digit training files and evaluated the algorithms on a test set comprising 160 continuous digit utterances.

B. Description of ASR Features

We train the speech recognizer on MFCCs with the first and second derivatives, with 16 states total for each digit model. 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 respective parameters in HTK to be both equal to 2 frames.

The feature extraction for the 23 spectral coefficients is done in MATLAB. We then find a sparse representation for these spectral coefficients with the optimization algorithms described above. From the denoised spectral coefficients, we reconstruct the 13 MFCC coefficients with the first and second

derivatives, which are then passed to the HTK implemented continuous digit recognizer for the speech recognition.

C. Details of Algorithms Implementation

Our optimization algorithms are implemented using MATLAB. The Group Sparse LASSO algorithm was implemented using SLEP 4.0 available at www.public.asu.edu/~jye02/Software/SLEP/ which provides an optimized MATLAB routine for solving the Sparse Group LASSO formulation.

For comparison with some widely known state-of-the-art denoising techniques, we have also included results obtained from Cepstral Mean Normalization (CMN) and the ETSI Advanced Front-end (ETSI AFE) [23].

For the sliding window implementation, we used an exemplar vector size of 805 and also a shift of 10 frames [4] for the Aurora 2.0 database and a shift of 3 frames for the Aurora 3.0 database.

D. Signal Reliability Mask

For our observed vector $\boldsymbol{f}$, there will be components that are more corrupted with noise as compared to the rest. Thus, if we estimate $\boldsymbol{x}$ from these unreliable components, the estimated $\widehat{\boldsymbol{x}}$ will not be very accurate for the reconstruction of $\boldsymbol{f}$. Thus, we employ a hard signal reliability mask [24] to denote reliable parts of the observed data. In particular, the mask matrix will be the same dimensions as that of the extracted feature matrix, with 1 to indicate a reliable entry and 0 to indicate a noisy entry. In this paper, we will use a mask which is estimated from the noisy speech data itself [25]. This involves getting 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. An estimate of the clean digit utterance is obtained by subtraction of the noise estimate from the noisy digit utterance.

After we have an indication of which component is reliable and which is not, we remove the components deemed unreliable from our observed vector $\tilde{\boldsymbol{f}}$. We also modify our dictionary $\boldsymbol{\Phi}$ to remove the unreliable rows corresponding to the unreliable components of $\tilde{\boldsymbol{f}}$. By varying the threshold SNR values we will be able to get more reliable estimates of $\boldsymbol{x}$. In this paper, like in [4], we adopt an SNR threshold of 20 dB.

E. Approaches for dictionary partitioning

There are many ways to partition the dictionary, of which we will explore two intuitive approaches:

- We partition the dictionary according to specific speaker identity. Here, we sort the columns of the dictionary to have the spectral segments of similar speakers grouped together, and then we do

chunking to ensure that each chunk will essentially have utterances from the same speaker. This is a knowledge-based approach for dictionary partitioning.

- Partition the dictionary according to proximity based on the L_2 distance. In this respect, we perform some form of clustering on the atoms of the dictionary based on the L_2 distance metric to form a pre-determined number of clusters, and then we run our grouped regularization technique based on these clusters. This is a data-driven approach for dictionary partitioning.

For the first method, we constructed the dictionary with 10 speaker groups.

For the second method, we perform the clustering using the popular K-means clustering algorithm [26]. In fact, there is a relationship between sparse representation and clustering. We can think of clustering as sparse representation in its extreme with one atom (the mean of the cluster) allowed in the representation, and the coefficient of the atom is 1. The K-means algorithm has also been generalized in the K-SVD algorithm [27] for designing overcomplete dictionaries for sparse represenation.

To decide on the number of clusters, we experimented on a smaller tuning set and discovered that, for 2-4 clusters, recognition accuracies were improving, while for more than 4 clusters the accuracies started to drop.

V. Results and Discussions

A. Evaluation Results for Aurora 2.0

We first evaluate the following algorithms on the SNR 5 dB noise corruption setting - Sparse Bayesian Learning (SBL), Least Angle Regression Implementation of the Elastic Net (LARS-EN), FISTA implementation of the Sparse Group LASSO (FISTA SGL), Group Elastic Net (Group EN), and lastly Group Sparse Bayesian Learning (Group SBL). We additionally evaluate LARS-EN, Group EN, and Group SBL for SNR -5 dB, SNR 0 dB, and SNR 10 dB.

Table I presents the results for optimized versions of the algorithms for the SNR 5 dB corruption. Table III presents the results for SNR -5 dB, 0 dB, 10 dB, and clean conditions.

[TABLE 1 about here.]

From Table I, we can see that when the dictionary is grouped appropriately, coupled with a suitable group sparse regularization algorithm, speech recognition performance can be greatly improved over the original base algorithm. In particular, for our evaluation of the SNR 5 dataset, we can see that the Group EN algorithm performs the best with a recognition accuracy of 67.78% with the speaker group clustering technique.

We see that the SBL algorithm did not perform as well as either the LASSO algorithm or the Elastic Net algorithm. However, with our Group SBL modification, it performs significantly better than the SBL algorithm. This can be explained by the fact that the original SBL formulation could be less adept at handling collinear dictionaries of spectral exemplars. In our formulation of the Group SBL algorithm, we are doing rank augmentation to our dictionary matrix (as evident from Equation (16)). Thus, when we are applying SBL as an intermediate step, the augmented dictionary has a higher chance of being of sufficient rank to result in better performance as demonstrated by our results in Table I and Table III.

Thus, we see that both our proposed algorithms (Group EN and Group SBL) with the appropriate groupings perform better than the Elastic Net and the SBL algorithm respectively. Moreover, the proposed algorithms are also comparable to the FISTA SGL algorithm and, in the case of the Group EN, outperforming it.

The results in Table I also show that clustering by knowledged-based speaker identity performs slightly better than the data-driven K-means clustering with the L_2 distance metric to partition the dictionary. Intuitively, similar spectral profiles will tend to cluster closer together with the L_2 distance. Thus, during the group sparse regularization process, irrelevant groups (consisting of spectral profiles which are further away in the L_2 sense) will be zeroed out, and the appropriate partition will have non-zero coefficients with sparsity enforced by our proposed algorithms. When we partition the dictionary based on speaker identity, we are retaining speaker characteristics within each partition, and distinct spectral profiles coming from different utterances will be present within a single partition. Thus, partitioning according to speaker identity can be likened to performing regularization with a series of more focused dictionaries, which explains why it is slightly outperforming the K-means clustering approach.

Another interpretation of K-means clustering is the following: K-means can be likened to an extreme version of sparse regularization as mentioned earlier. Thus, the additional group sparse regularization step can be viewed as a refinement step of the K-means, which has been empirically demonstrated to yield good recognition rates given the appropriate refinement algorithm in Tables I and III.

In fact, our algorithms can be likened to a subset of the feature compression algorithm in the ETSI AFE [23]. Specifically, the dictionary learning algorithms is analogous to the vector quantization step in the compression algorithm pipeline of ETSI AFE.

[Fig. 2 about here.]

Regularization in the spectral domain amounts to spectral denoising [4]. From the spectral plots, we can see that LARS-EN is improving upon the noisy version with successful removal of portions of the

noise artifacts, and the Group EN algorithm further improves upon the LARS-EN algorithm by closer resemblance to the original clean spectrum, as illustrated in the example of Fig. 2.

B. Runtimes

Table II presents runtime results for our algorithms and the Sparse Group LASSO algorithm.

[TABLE 2 about here.]

We can see that FISTA SGL runs the fastest followed closely by Group EN and, lastly, by Group SBL. To ensure that we are comparing the algorithms fairly, the algorithms are ensured to have converged sufficiently and are evaluated on the same platform (Core 2 Quad Processor with 8 gigabytes of RAM). All algorithms are evaluated in MATLAB.

In our group regularization formulations, we can see that we are retaining the complexity advantages of the base algorithms. In particular, if we cap the iterations to a limit, the complexity is the same as that of the base algorithms.

It has been proven that the Iterative Shrinkage Thresolding Algorithm (ISTA) has a sublinear convergence ($O(\frac{1}{i})$ where i is the iteration) and FISTA further improves on it by a quadratic factor ($O(\frac{1}{i^2})$) [14]. In this paper, we have experimentally verified that our Group EN formulation runs at comparable speed as the FISTA implementation.

C. Aurora 2.0 Results for other SNR levels

[TABLE 3 about here.]

As evident from the results in Table III, we see that the Group EN algorithm is consistently better than the Elastic Net under different levels of noise corruption, which shows that grouping helps in improving speech recognition rate.

An important aspect of denoising algorithms is performance under clean conditions. We evaluated the proposed algorithms in clean conditions, and results show that the Group EN algorithm suffers from some amount of deterioration, while the Group SBL exhibits good robustness in clean conditions. For good reconstruction under clean conditions, sparsity is important and, since the Group SBL algorithm is enforcing sparsity differently from the Group EN, we see that the better performance of the Group SBL algorithm when compared to the Group EN algorithm under cleaner conditions could imply that the Group SBL way of enforcing sparsity is more efficient under cleaner conditions.

D. Evaluation of Algorithms on Real Noisy Data

We additionally evaluate our algorithms on the Aurora 3.0 Italian database. Our test set is comprised of 160 continuous digit utterances, randomly selected from the following noise types: low speed rough road, town traffic, stop motor running, and high speed good road. Table IV shows recognition results after denoised by Lars-EN, Group EN, Group SBL, the ETSI Advanced Front-end, and lastly Cepstral Mean Normalization.

[TABLE 4 about here.]

On this realistic dataset, we see that the Group EN is the best performing, followed very closely by the ETSI AFE and the Group SBL algorithm. We see that, in general, our proposed dictionary-based sparse regularization algorithms are comparable with state-of-the-art techniques and, in particular for the Group EN algorithm, we are slightly outperforming ETSI AFE. For Aurora 2.0, since the dataset is synthetically corrupted with additive noise, we can expect to observe some algorithms performing better than others due to their handling of additive noise better. However, in the presence of the more complicated real noisy conditions, the robustness of our proposed algorithms is made apparent.

VI. Conclusion and Final Remarks

In this paper, we introduced two novel variations of group sparse regularization techniques: the Group Elastic Net algorithm and the Group Sparse Bayesian Learning algorithm. We see that the Group EN algorithm is the better performing of both algorithms for the Aurora 2.0 and 3.0 dataset. We also see that the Group SBL algorithm exhibited least deterioration with clean data and improved greatly upon the performance of SBL. Moreover, on the Aurora 3.0 database, the Group SBL algorithm exhibited greater robustness compared to Elastic Net under the more practical real noise condition. In terms of runtime, we see that the Group EN and Group SBL algorithms are comparable with FISTA SGL. We also see that the results obtained from our algorithms are comparable with the ETSI AFE. We believe that further improvements can be achieved with an adaptive dictionary update approach. In this work, the overcomplete dictionary is formed by a simple strategy. In adaptive dictionaries, essentially we would update the dictionary as we perform regularization. This can ensure that only representative spectral segments are retained in the dictionary, and irrelevant spectral segments are eliminated.

The paper also explored two intuitive strategies for partitioning the dictionary of exemplars, namely by K-means clustering with the L_2 distance metric and also by speaker identity. We find that the speaker clustering strategy performs better than the clustering with K-means using the L_2 distance metric.

Future work in terms of investigating better partitioning strategies can be directed towards finding meaningful hybrids between data-driven approaches and knowledge-based approaches.

Another interesting extension will be to expand our framework to be able to do speaker identification. In this work, the speakers in the dictionary do not generally overlap with the speakers in the test set, so speaker identification is not possible within the current framework. However, due to the fact that the partitioning is doing some form of discrimination, our techniques could potentially be adapted for this task.

APPENDIX
DERIVATION OF PROPERTY 1

From Equation (20) we have

$$P(\gamma_j^{-1}) = \frac{1}{\Gamma(m_\gamma)} {n_\gamma}^{m_\gamma} \gamma_j^{1-m_\gamma} e^{-n_\gamma \gamma_j^{-1}}$$

where $\Gamma(.)$ is the Gamma function. Thus,

$$\begin{aligned}
p(\boldsymbol{x}_{ij}'') &= \int p(\boldsymbol{x}_{ij}''|\gamma_j) p(\gamma_j) d\gamma_j^{-1} \\
&= \int \frac{1}{(2\pi\gamma_j)^{\frac{1}{2}}} e^{-\frac{{\boldsymbol{x}_{ij}''}^2}{2\gamma_j}} \frac{\Gamma(m_\gamma)}{{n_\gamma}^{m_\gamma} \gamma_j^{1-m_\gamma} e^{-n_\gamma \gamma_j^{-1}}} d\gamma_j^{-1} \\
&\propto \int \gamma_j^{\frac{1}{2}-m_\gamma} e^{-\frac{\frac{{\boldsymbol{x}_{ij}''}^2}{2}+n_\gamma}{\gamma_j}} d\gamma_j^{-1} \\
&= \int \left(\frac{{\boldsymbol{x}_{ij}''}^2}{2} + n_\gamma\right)^{-m_\gamma-\frac{1}{2}} (\gamma_j')^{m_\gamma-\frac{1}{2}} e^{-\gamma_j'} d\gamma_j' \\
&= \left(\frac{{\boldsymbol{x}_{ij}''}^2}{2} + n_\gamma\right)^{-m_\gamma-\frac{1}{2}} \Gamma\left(m_\gamma + \frac{1}{2}\right)
\end{aligned}$$

Where $\gamma_j' = \frac{\frac{{\boldsymbol{x}_{ij}''}^2}{2}+n_\gamma}{\gamma_j}$. Thus we have:

$$p(\boldsymbol{x}_{ij}'') \propto \left(\frac{{\boldsymbol{x}_{ij}''}^2}{2} + n_\gamma\right)^{-m_\gamma-\frac{1}{2}}$$

As $m_\gamma, n_\gamma \to 0$, we have $p(\boldsymbol{x}_{ij}'') \to c' \frac{1}{|\boldsymbol{x}_{ij}''|}$ where c' is some constant.

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.

REFERENCES

[1] 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.

[2] J. Gemmeke and B. Cranen, "Missing data imputation using compressive sensing techniques for connected digit recognition," *Proceedings of the 16th international conference on Digital Signal Processing*, pp. 37–44, 2009.

[3] 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.

[4] Q. F. Tan, P. G. Georgiou, and S. S. Narayanan, "Enhanced Sparse Imputation Techniques for a Robust Speech Recognition Front-End," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 8, pp. 2418 – 2429, 2011.

[5] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, pp. 600–616, 1997.

[6] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] 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.

[8] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[10] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[11] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 1, pp. 53–71, 2008.

[12] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *preprint*, 2010.

[13] J. Liu, S. Ji, and J. Ye, "SLEP: Sparse Learning with Efficient Projections," *Arizona State University*, 2009.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[15] 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.

[16] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[17] H. Zou, "The adaptive LASSO and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[18] P. Zhao and B. Yu, "On model selection consistency of LASSO," *The Journal of Machine Learning Research*, vol. 7, p. 2563, 2006.

[19] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[20] M. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1, no. 3, 2003.

[21] A. Faul and M. Tipping, "Analysis of Sparse Bayesian learning," in *NIPS*, vol. 14, 2002.

[22] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," 2000.

[23] ETSI ES 201 108, "v1.1.3 Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Frontend feature extraction algorithm; Compression algorithms," September 2003.

[24] 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.

[25] 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.

[26] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006.

[27] M. Aharon, M. Elad, A. Bruckstein, and K. Yana, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

Qun Feng Tan received the B.S.E. degree in electrical engineering with a minor in mathematics from the University of Michigan, Ann Arbor, in 2006, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2008. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Southern California, Los Angeles, under the Provost Fellowship. His research interest lies in signal processing and pattern recognition with applications to speech processing. Other interests include recreational mathematics and number theory.

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 the Acoustical Society of America, the Institute of Electrical and Electronics Engineers, and the American Association for the Advancement of Science. He is a member of Tau Beta Pi, Phi Kappa Phi and Eta Kappa Nu. He is 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 previously an Associate Editor of the IEEE Transactions of Speech and Audio Processing (2000-04) and the IEEE Signal Processing Magazine (2005-2008). He holds positions on the Speech Communication and Acoustic Standards committees of the Acoustical Society of America and the Advisory Council of the International Speech Communication Association.

Shri Narayanan has received a number of professional honors including an a 2005 Best Paper award (with Alex Potamianos) and a 2009 Best Paper Award (with Chul Min Lee) from the IEEE Signal Processing society and was selected as an IEEE Signal Processing Society Distinguished Lecturer for 2010-2011. Papers co-authored with his students have won awards at Interspeech 2011 Speaker State Challenge, InterSpeech 2010, InterSpeech 2009-Emotion Challenge, IEEE DCOSS 2009, IEEE MMSP 2007, IEEE MMSP 2006, ICASSP 2005 and ICSLP 2002. He has published over 450 papers and has 12 granted U.S. patents.

List of Figures

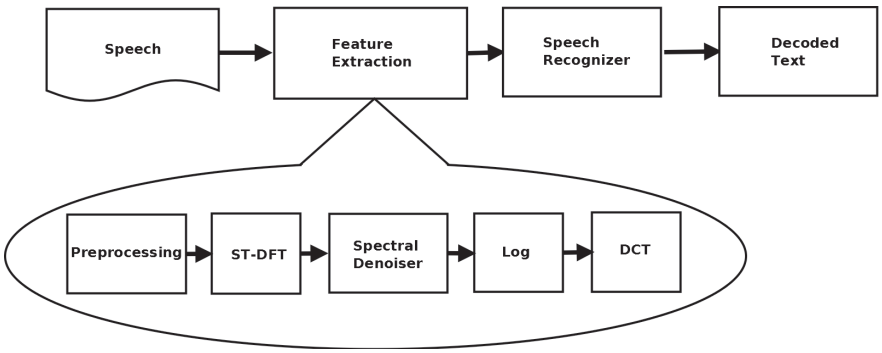

Fig. 1. Block diagram illustrating a typical speech recognition system but enhanced with our new front-end. The "Spectral Denoiser" module is an extra module which we introduced into the feature extraction flow that utilizes the group sparse regularization techniques with appropriate dictionary partitioning for better recognition accuracies.

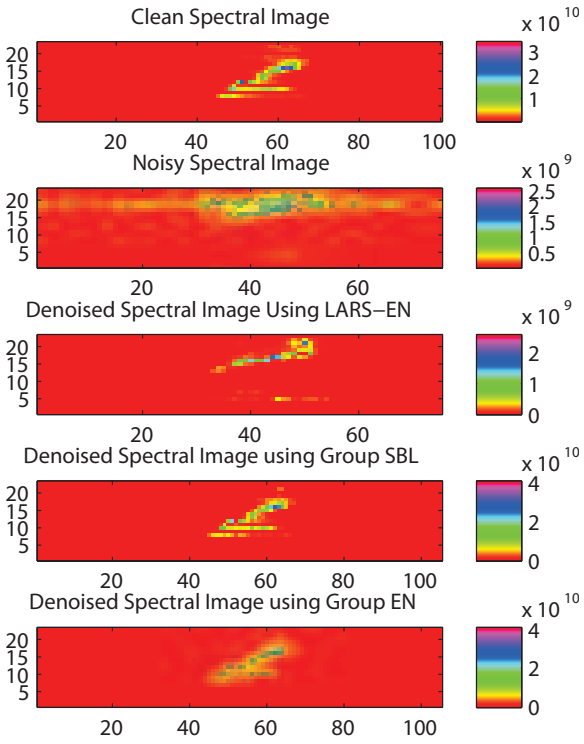

Fig. 2. The diagram above shows the spectral plots of a particular utterance. As we can see, LARS-EN improves greatly upon the noisy version with much of the noise artifacts eradicated, and the Group EN algorithm further improves upon the LARS-EN algorithm by a closer resemblance to the original clean spectrum.

LIST OF TABLES

TABLE I
SPECTRAL DENOISING RESULTS FOR THE AURORA 2.0 DATABASE.

Algorithm	Grouping	Accuracy (%)
Before denoising	NA	43.82
SBL	NA	39.70
LASSO	NA	64.24
LARS-EN	NA	66.36
Group EN	Speaker	67.78
Group EN	L_2	67.42
FISTA SGL	Speaker	56.81
FISTA SGL	L_2	53.38
Group SBL	Speaker	55.80
Group SBL	L_2	48.16
ETSI AFE	NA	72.67
CMN	NA	55.74

TABLE II
AVERAGE RUNTIME RESULTS FOR THE GROUP SPARSE REGULARIZATION ALGORITHMS FOR SNR 5 dB CORRUPTION

Algorithm	Runtime (secs/optimization)
FISTA SGL	0.06
Group EN	0.07
Group SBL	0.11

TABLE III
RESULTS FOR SNR -5 dB, SNR 0 dB, SNR 10 dB CORRUPTION AND CLEAN CONDITIONS FOR AURORA 2.0

Algorithm	Accuracy (%)
SNR -5 dB	
Before denoising	7.89
LARS-EN	31.04
Group EN	31.31
Group SBL	26.15
SNR 0 dB	
Before denoising	18.59
LARS-EN	46.83
Group EN	47.80
Group SBL	38.10
SNR 10 dB	
Before denoising	68.89
LARS-EN	82.16
Group EN	83.24
Group SBL	70.86
Clean Conditions	
Clean	98.99
Group EN	97.78
Group SBL	98.98

TABLE IV
RESULTS FOR AURORA 3.0 NOISY DATASET

Algorithm	Accuracy (%)
Before denoising	62.87
LARS-EN	75.26
Group EN	77.50
Group SBL	76.46
ETSI AFE	77.21
CMN	67.09