Sparse Modeling of Neural Network Posterior Probabilities for Exemplar-based Speech Recognition

Pranay Dighe[a,b,*], Afsaneh Asaei[a], Hervé Bourlard[a,b]

{*pranay.dighe,afsaneh.asaei,herve.bourlard*}*@idiap.ch*

[a]*Idiap Research Institute, Martigny, Switzerland*
[b]*Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*

Abstract

In this paper, a compressive sensing (CS) perspective to exemplar-based speech processing is proposed. Relying on an analytical relationship between CS formulation and statistical speech recognition (Hidden Markov Models - HMM), the automatic speech recognition (ASR) problem is cast as recovery of high-dimensional sparse word representation from the observed low-dimensional acoustic features. The acoustic features are exemplars obtained from (deep) neural network sub-word conditional posterior probabilities. Low-dimensional word manifolds are learned using these sub-word posterior exemplars and exploited to construct a linguistic dictionary for sparse representation of word posteriors. Dictionary learning has been found to be a principled way to alleviate the need of having huge collection of exemplars as required in conventional exemplar-based approaches, while still improving the performance. Context appending and collaborative hierarchical sparsity are used to exploit the sequential and group structure underlying word sparse representation. This formulation leads to a posterior-based sparse modeling approach to speech recognition. The potential of the proposed approach is demonstrated on isolated word (Phonebook corpus) and continuous speech (Numbers corpus) recognition tasks.

Keywords: Automatic speech recognition, Deep neural network posterior features, Compressive sensing, Sparse word posterior probabilities, Dictionary learning, Sparse modeling

1. Introduction

Hidden Markov model (HMM) based modeling and template (exemplar) based techniques are the two main approaches towards automatic speech recognition (ASR). In the last three decades, HMM-based approaches have been dominant because of their flexibility and their ability to be trained and generalized to unseen data. In comparision, exemplar based techniques use labeled

*Corresponding author

speech segments (called *exemplars*) directly for speech recognition, without a model learning step as done in HMM based systems. Assuming an "infinite" amount of such exemplars, as well as the "right" representation space and the "right" distance measure, "optimal" recognizers could be sought in theory (Devijver and Kittler, 1982). The trade-off is that such a system will have a huge space and time complexity. However, with the ever increasing amount of training data, as well as the growing computational and memory resources, the potential of exemplar-based approaches is currently being explored extensively (Sainath et al., 2012, 2011; Gemmeke et al., 2011; De Wachter et al., 2007).

One of the emerging approaches in exemplar-based ASR is exemplar-based sparse representation in which a test speech segment is expressed as a sparse linear combination of the exemplars in the training dataset. Thus, a large collection of exemplars is used in practice to capture all possible variability in the data. The core assumption of this approach is that any possible realization of the data in the test set lies in a vector space spanned by a sparse selection of exemplars already seen in the training set or/i.e. the exemplars live in low-dimensional manifolds. This assumption hints towards possibilities of posing the exemplar-based sparse representation problem as a typical compressive sensing problem in which the goal is to find an over-complete set of basis vectors (termed as *Dictionary* in CS), a sparse linear combination of which can be used to generate all the data points. When seen through this perspective, the exemplar-based sparse representation task becomes exposed to the well studied theory and techniques associated with compressive sensing – namely dictionary learning and sparse recovery. Further, if the dictionaries are designed in a particular manner, the compressive sensing procedure can be given a very intuitive probabilistic interpretation in terms of the speech recognition theory (as we shall see in Section 3). This paper is an attempt towards exploring all these possibilities to devise a novel framework for ASR based on compressive sensing.

It has been shown previously that the acoustic feature space lies on one or more low-dimensional manifolds (Stevens, 1998; Jansen and Niyogi, 2006). In case of speech exemplars, these acoustic features are derived from speech segments which actually represent sub-word units e.g. syllable, phone or even sub-phones. The occurrence of these units leads to a sparse event for example in the context of high-dimensional word representation space. This enables us to cast the speech recognition problem as reconstruction of a high-dimensional sparse word representation from the low-dimensional sub-word acoustic exemplars.

Another inspiration for the CS based approach comes from some issues faced by conventional exemplar-based sparse representation systems. Firstly, it has been reported (section 5.3 in Gemmeke et al. (2009)) that increasing the size of exemplar collection improves the ASR performance only upto a certain limit, after which improvement in ASR performance is sub-linear. At certain point, the additional information brought in the collection by new exemplars is insignificant as they are close to existing exemplars in the collection. This suggests the need for a better procedure to find a limited size collection of exemplars that can be used for sparse representation of data. In this paper, we propose to

use dictionary learning in order to exactly address this need by finding an over-complete set of basis vectors which spans the vector space. We demonstrate experimentally (section 5) that the dictionary so learned has a cardinality far smaller than the size of collection of all exemplars of the training data, but still improves characterisation of the vector space as compared to the collection of exemplars. This observation confirms that dictionary learning is a more efficient way for sparse representation than exemplar collection.

Secondly, the existing exemplar-based sparse methods do not have a specific built-in mechanism for exploiting the temporal relationships across consecutive exemplars. Context-appended frames have been used in (Gemmeke et al., 2011, 2009) to deal with this issue. In our approach, we propose to utilise techniques like collaborative group sparsity (Sprechmann et al., 2011) to seek collaboration among sparse coding of consecutive exemplars. Our idea is to demonstrate how sparse modeling can lead to a different paradigm in pattern matching for speech recognition by offering a hierarchical structure instead of enforcing the Markovian inter-dependency.

To the best of our knowledge, all of the proposed exemplar-based sparse approaches use spectral-based features (Sainath et al., 2011; Gemmeke et al., 2011, 2009; Sainath et al., 2010). In contrast to the state-of-the-art, we derive a probabilistic interpretation of the problem (section 3) that leads to a requirement of posterior probability-based features in place of spectral features in our case. We use phone conditional posterior probabilities as exemplars to build the dictionaries, and sparse coding is used as a method of recovering the sparse word posterior probabilities. In our earlier study, posterior features have been shown to yield promising results in exemplar-based speech recognition (Bahaadini et al., 2014); where using the link to the statistical speech recognition formalism, new derivation of super/sub-phone posterior features are developed for exemplar-based sparse representation. Successful reconstruction of the sparse word posterior representation requires (1) learning a dictionary to characterize the manifold of word sub-spaces and (2) devising an effective sparse recovery procedure to estimate the word posterior probabilities from the compressive sub-word observations. These two subjects are thoroughly studied in this work.

The remainder of this paper is organized as follows. In Section 2, we provide a background on posterior features, compressive sensing and sparse modeling. Section 3 presents a novel compressive sensing perspective towards posterior based sparse modeling. A speech recognition framework based on the CS formulation is presented in Section 4 where we also briefly dicsuss its links to HMM and DTW techniques. Section 5 presents the details of the experiments and the conclusions are drawn in Section 6 along with the directions for future research.

2. Background

In this section, we discuss some background information on the use of posterior features in ASR, as well as compressive sensing and sparse signal reconstruction. The notations used in this paper are as follows

- $q_k, \quad \forall k \in \{1, \ldots, K\}$: sub-word units.
- $w_l, \quad \forall l \in \{1, \ldots, L\}$: word units.
- $x_t, \quad \forall t \in \{1, \ldots, T\} \in \mathbb{R}^d$: spectral speech features at time t.
- $\mathbf{X} = [x_1 \ldots x_T] \in \mathbb{R}^{d \times T}$: sequence of spectral speech features x_t.
- $z_t = [p(q_1|x_t) \ldots p(q_K|x_t)]^\top \in \mathbb{R}^K$: posterior probability (acoustic) features; $.^\top$ stands for transpose.
- $\mathbf{Z} = [z_1 \ldots z_T] \in \mathbb{R}^{K \times T}$: sequence of posterior features $z_t, \forall t \in \{1, \ldots, T\}$.
- $\mathbf{D} = [d_1 \ldots d_L] \in \mathbb{R}^{K \times L}$: dictionary of exemplars where d_l denotes each column/atom of the dictionary.
- $\alpha_t \in \mathbb{R}^L$: sparse reconstructed vector corresponding to posterior vector z_t.
- $\mathbf{A} = [\alpha_1 \ldots \alpha_T] \in \mathbb{R}^{L \times T}$: sparse reconstructed matrix corresponding to input posterior matrix $\mathbf{Z}$.

2.1. Posterior Features

The setup for extracting the phone posterior features is illustrated in Figure 1 (Aradilla et al., 2009). The spectral features, comprised of 13 MFCC cepstral coefficients with their first and second order derivatives, are computed over a sliding window of 25ms with a shift of 10ms. A multilayer perceptron (MLP) or (deep) neural network (DNN) is used to take in a context of spectral features as inputs and generate the phone posterior probabilities. The output probability vector can be directly used as acoustic features for speech recognition, thus referred to as the posterior features. In our case, a context of 4 frames is used as input for the neural network. The output layer includes an additional (phone) unit for representing the silence/pause along with the other phones.

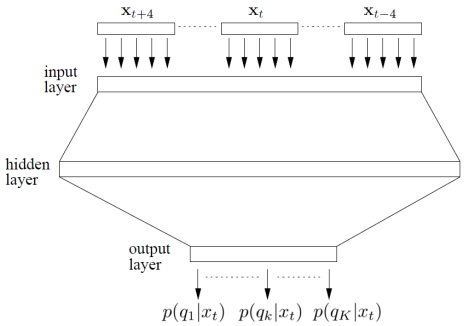

Figure 1: Posterior features are extracted using a neural network taking the spectral features as input.

Obviously and as confirmed in (Asaei et al., 2010), the posterior features are sparse whereas the conventional probabilistic modeling framework for speech

recognition is not designed to handle sparse features. The recently developed Kullback-Leibler HMM (KL-HMM) acoustic modeling framework have been shown to be more effective to exploit the characteristics of posterior features (Aradilla et al., 2008). These features are more robust to speaker and environmental availabilities, thus outperform spectral features in realistic speech recognitions tasks.

2.2. Compressive Sensing and Sparse Modeling

Compressive sensing relies on sparse representation to reconstruct a high-dimensional data using very few linear non-adaptive observations. A data representation $\alpha \in \mathbb{R}^M$ is N-sparse if only $N \ll M$ entries of α have nonzero values. We call the set of indices corresponding to the non-zero entries as the support of α. The CS theory asserts that only $K = O(N \log(M/N))$ linear measurements, $z \in \mathbb{R}^K$ obtained as

$$z = \mathbf{D}\,\alpha \tag{1}$$

suffice to reconstruct α, where $K \ll M$ and $\mathbf{D} \in \mathbb{R}^{K \times M}$ is a measurement matrix which can also be interpreted as an over-complete dictionary designed/learned for sparse representation of α.

A sufficient but not necessary condition on $\mathbf{D}$ to recover the sparse data representation coefficients is that all pairwise distances between N-sparse signals must be well preserved in the observation space or equivalently all subsets of N columns taken from the dictionary are in fact nearly orthogonal. While there are infinitely many solutions to equation (1), relying on the two ingredients (1) sparse representation and (2) incoherent measurement, CS guarantees to circumvent the ill-posedness of the problem and recover the N-sparse data stably from the compressed (low-dimensional) observations through efficient optimization algorithms which search for the sparsest representation that agrees with those observations.

Given an observation vector z, and an over-complete dictionary matrix $\mathbf{D}$, the sparse representation α is obtained by the optimization problem stated as:

$$\min_{\alpha} \|\alpha\|_0 \quad \text{subject to} \quad z = \mathbf{D}\,\alpha \tag{2}$$

where the counting function $\|.\|_0 : \mathbb{R}^M \longrightarrow \mathbb{N}$ returns the number of non-zero components in its argument. The non-convex objective of (2) is often relaxed to ℓ_1-norm optimization which can be solved in polynomial time; the ℓ_1 norm, $\|\alpha\|_1$ is defined as sum of the absolute values of the components of α. Further developments consider alternative data reconstruction metrics tailored for a specific application such as classification. Recent advances in CS exploit the inter-dependency structure underlying the support of the sparse coefficients in recovery algorithms to reduce the number of required observations and to better differentiate true coefficients from recovery artifacts which leads to a more robust and efficient recovery (Asaei et al., 2011).

In the following subsections, we will briefly discuss the methodologies to learn the dictionary $\mathbf{D}$ and solving the sparse recovery optimization expressed in (2).

2.2.1. Dictionary Learning

The goal of dictionary learning is to optimize an overcomplete basis set such that the training feature vectors can be characterized as a sparse linear superposition of the basis vectors. This approach assumes that the training data live in a low-dimensional (non-Euclidean) space that can be modeled as an union of sub-spaces. An overcomplete dictionary, which has more atoms than the dimensions of the subspaces, attempts not only to capture the broad range of variability that the data can exhibit, but also helps in decompressing the initial compact feature-space to a high dimensional sparse space where discrimination between various data phenomena becomes easier. This favorable property of dictionary learning is exactly what we need for the task of speech recognition where the variability comes from countless sources like gender, age, accent, surroundings etc. The other requirement for our task is the efficient scalability of the system to larger datasets. With availability of huge datasets, an algorithm which can utilize all the available knowledge will be preferred.

Given a training set of features $\mathbf{Z} = [z_1, ..., z_T] \in \mathbb{R}^{K \times T}$, a dictionary $\mathbf{D} \in \mathbb{R}^{K \times M}$ and sparse representation $\mathbf{A} = [\alpha_1, ..., \alpha_T]$ for $\mathbf{Z}$; the ℓ_1-based sparse recovery based objective function for classical dictionary learning techniques is defined as

$$\arg\min_{\mathbf{D},\mathbf{A}} \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{2} \|z_t - \mathbf{D}\,\alpha_t\|_2^2 + \lambda \|\alpha_t\|_1 \right) \tag{3}$$

where λ is the regularization parameter. The first term in this expression, quantified the *reconstruction error* whereas the second term controls the sparsity of α_t. The joint optimization of this objective function with respect to both $\mathbf{D}$ and α_t simultaneously is non-convex, it can be solved as a convex objective by optimizing for one while keeping the other fixed. In this paper, we study the performance of two main approaches to dictionary learning using the posterior-based exemplars. The concept of these techniques are briefly summarized here.

One of the prominent algorithms for dictionary learning is K-SVD algorithm developed by Aharon and Elad (Aharon et al., 2006). It roughly generalizes the idea of k-means clustering to the task of dictionary learning. The dictionary is learned atom by atom using the singular value decomposition (SVD) to minimize the quadratic reconstruction error associated to each atom. To that end, the dictionary is initialized and the sparse representation of the posterior features are obtained. Then, a residual error $\mathbf{E}_j$ is defined when atom d_j is removed along with its corresponding coefficients, i.e. j^{th} row of $\mathbf{A}$ which is denoted as a_T^j. Hence, each dictionary atom and its associated sparse coefficients is updated though

$$d_j^{\text{new}}, a_T^{j\,\text{new}} = \arg\min_{d_j, a_T^j} \left\| \mathbf{E}_j - d_j\, a_T^j \right\|_{\text{F}}^2. \tag{4}$$

The SVD is used to find the closest rank-1 decomposition of $\mathbf{E}_j$ to update d_j and a_T^j. This procedure is repeated for all atoms of the dictionary. To ensure the sparsity in $\mathbf{A}$, only those columns of $\mathbf{E}_j$ are used for decomposition that correspond to z_t's in $\mathbf{Z}$ which use the atom d_j in their sparse representation.

Another important algorithm is a fast online optimization proposed by (Mairal et al., 2010) for learning dictionaries based on stochastic approximations. The algorithm basically alternates between a step of sparse coding for the current training feature z_t and then optimizes the previous estimate of dictionary $\mathbf{D}^{(t-1)}$ to determine the new estimate $\mathbf{D}^{(t)}$ using stochastic gradient descent. The algorithm has been shortly summarized in Algorithm 1.

Algorithm 1 Online Dictionary Learning

Require: : $\mathbf{Z} = [z_1, ..., z_T] \in \mathbb{R}^{K \times T}, \lambda \in \mathbb{R}$: regularization parameter, initial estimate for dictionary $\mathbf{D}^{(0)} \in \mathbb{R}^{K \times M}$

1: **for** $t = 1$ `to` T **do**
2: Sparse Coding of z_t to determine α_t:
$\alpha_t = \arg\min_\alpha \left\{ \frac{1}{2} \|z_t - \mathbf{D}^{(t-1)}\alpha\|_2^2 + \lambda \|\alpha\|_1 \right\}$
3: Updating $\mathbf{D}^{(t)}$ with $\mathbf{D}^{(t-1)}$ as warm restart:
$\mathbf{D}^{(t)} = \arg\min_{\mathbf{D}} \left\{ \frac{1}{t} \sum_{i=1}^{t} (\frac{1}{2} \|z_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1) \right\}$
4: **end for**
5: **return** $\mathbf{D}^{(T)}$

2.2.2. Sparse Recovery

The computational methods to solve the sparse recovery problem expressed in (2) are reviewed in (Tropp and Wright, 2010). In this paper, we study the performance of two main approaches to sparse reconstruction of posterior features. The concept of these techniques are briefly summarized here.

One of the major algorithmic approaches to sparse recovery relies on greedy pursuit of basis vectors referred to as the orthogonal matching pursuit (OMP). OMP is an iterative greedy method which finds a sparse solution for (2) by repeatedly identifying one or more atoms of the dictionary that yield the highest improvement in minimization of reconstruction error (Davis et al., 1997; Tropp and Wright, 2010). A major advantage of this approach is that it does not need to relax the ℓ_0 norm criterion, so one can control the sparsity as required. The stopping criterion can be chosen by fixing the number of atoms.

An alternative to the greedy sparse recovery is to relax the problem stated in (2) as a convex objective by replacing the $\|.\|_0$-norm with $\|.\|_1$-norm which is referred to as the least absolute shrinkage and selection operator (Lasso) (Tibshirani, 1996). It is known that relaxing the combinatorial problem of ℓ_0 norm to ℓ_1 constraint leads to (equivalent) sparse solutions for α. While the former is NP-hard, the relaxed formulation admits efficient polynomial time algorithms. Furthermore, the solutions of l_1-norm minimization is less sensitive to noise. The standard Lasso problem can be solved by various convex optimization techniques. One of the efficient and computationally fast techniques is LARS implementation (Efron et al., 2004) which we consider for the present study. We will explain in Section 3 that the posterior feature space can be very elegantly posed into formulations which lead to hierarchical group sparsity. Thus, we can leverage the variants of Lasso which specifically deal with such structured

sparsity. The collaborative hierarchical lasso optimization is considered for our studies, thus we briefly state its objective for structured sparse recovery.

In hierarchical group Lasso, sparsity is sought at a group level as well as for the individual atoms of the dictionary. A set of groups $\mathcal{G} = [G_1, \cdots, G_L]$ is simply a partitioning over the dictionary. The collaborative hierarchical group Lasso (C-HiLasso) is developed in (Sprechmann et al., 2011). This algorithm enables us to incorporate the dependency among a sequence of posterior vectors z_t's by defining an objective function for collaboration. By collaboration, we mean that the collection of z_t's share the same non-zero components in α_t's. Thus, the collaborative group problem is formulated as (Sprechmann et al., 2011)

$$\min_{\alpha} \frac{1}{2}\|\mathbf{Z} - \mathbf{D}\mathbf{A}\|_{\mathrm{F}}^2 + \lambda_2 \psi_{\mathcal{G}}(\mathbf{A}) + \lambda_1 \sum_{t=1}^{T} \|\alpha_t\|_1 \tag{5}$$

where $\psi_{\mathcal{G}}$ is group Lasso regularizer defined as $\psi_{\mathcal{G}}(\mathbf{A}) = \sum_{G \in \mathcal{G}} \|\mathbf{A}^G\|_F$ and $\mathbf{A}^G$ is the submatrix formed by all the rows belonging to group G.

It may be noted that the sparse recovery algorithms stated above are devised for Euclidean-norm quantification of data fidelity. The earlier studies on exemplar-based sparse representation exploits the generalized Kullback Liebler (GKL) divergence defined as

$$\mathrm{GKL}(z|\hat{z}) = \sum_{k=1}^{K} z(k) \log \frac{z(k)}{\hat{z}(k)} - z(k) + \hat{z}(k) \tag{6}$$

for sparse recovery (Gemmeke et al., 2011). This method is also considered as a benchmark in our experimental analysis in Section 5.

3. Compressive Sensing Perspective to Posterior-based Sparse Modeling

Speech recognition aims to recover the sequence of words from the observed acoustic features. The space of sub-word observation is low-dimensional (e.g. $\mathbb{R}^{K \times T}$) whereas the word transcription requires reconstructing a high-dimensional representation (e.g. $\mathbb{R}^{L \times T}$, $L \gg K$). The key idea is that the representation of linguistic information in the form of words for a given utterance is highly sparse. Hence, we propose to cast the speech recognition problem as sparse reconstruction of word representation given the compressed (low-dimensional) acoustic observation. The dictionary for sparse representation is formed from the sub-word exemplars to characterize the projection of the word sub-spaces to the space of input posterior features.

To state it more precisely, we define the set of acoustic units as $\{q_k\}_{k=1}^{K}$. Given an input feature vector x_t at time t, the posterior probability $p(q_k|x_t)$, is estimated at the MLP/DNN output where q_k is associated with a phone (Section 2.1). The set of phone posteriors correspond to the word level[1] posterior

[1] In principle, the hidden variable can also correspond to sub-phone units such as HMM-

probabilities through marginalization over L hidden variables w_l as follows:

$$p(q_k|x_t) = \sum_{l=1}^{L} p(q_k, w_l|x_t) = \sum_{l=1}^{L} p(q_k|w_l, x_t)p(w_l|x_t) = \sum_{l=1}^{L} p(q_k|w_l)p(w_l|x_t), \tag{7}$$

where the last equality holds due to conditional independence of the acoustic observation and input speech given a super-phone lexical unit such as word.

Considering the observation z_t consisting of the phone posterior features as $z_t = [p(q_1|x_t), \cdots, p(q_K|x_t)]^\top$, an over-complete dictionary $\mathbf{D}$ can be constructed such that the atoms are exemplars obtained by conditioning the phone posteriors on a different linguistic unit w_l. Designing the dictionary in this manner, we can now exploit (2) and (7) to define the sparse posterior-based representation α_t as $\alpha_t = [p(w_1|x_t), \cdots, p(w_L|x_t)]^\top$. Equation (1) now takes the following form:

$$\underbrace{\begin{bmatrix} p(q_1|x_t) \\ p(q_2|x_t) \\ \vdots \\ p(q_K|x_t) \end{bmatrix}}_{z_t} = \underbrace{\begin{bmatrix} p(q_1|w_1) & \cdots & p(q_1|w_l) & \cdots & p(q_1|w_L) \\ p(q_2|w_1) & \cdots & p(q_2|w_l) & \cdots & p(q_2|w_L) \\ \vdots & & \vdots & & \\ p(q_K|w_1) & \cdots & p(q_K|w_l) & \cdots & p(q_K|w_L) \end{bmatrix}}_{\text{Dictionary: } \mathbf{D}=[d_1 \ldots d_l \ldots d_L]} \times \underbrace{\begin{bmatrix} p(w_1|x_t) \\ \vdots \\ p(w_l|x_t) \\ \vdots \\ p(w_L|x_t) \end{bmatrix}}_{\alpha_t} \tag{8}$$

where

$$d_l = [p(q_1|w_l) \cdots p(q_k|w_l) \cdots p(q_K|w_l)]^\top$$

Based on (7), if z_t and $\mathbf{D}$ are composed of posterior features, α_t is also a posterior vector. The hidden variable w_l need not necessarily be associated with a word only; it can be interpreted as any other linguistic unit. In fact, equation (8) demonstrates how an acoustic feature vector $z_t = [p(q_1|x_t), \cdots, p(q_K|x_t)]^\top$ can be used for recovering the sparse posterior probabilities in a different linguistically defined space, $\alpha_t = [p(w_1|x_t), \cdots, p(w_L|x_t)]^\top$, using a dictionary $\mathbf{D}$ constructed from *appropriate* exemplars representative of the associated labels or hidden variables. In Figure 2, we represent this relation using a graphical model and compare it to the conventional acoustic modeling.

In practice, construction of the dictionary as described in (8) requires modeling the sub-spaces of each word using the acoustic features in terms of phone posterior probabilities. To characterize the posterior probabilities of each word, we learn word-specific dictionaries such that each column of the dictionary in (8),

states (Bahaadini et al., 2014). In this study, we consider the super-level linguistic units.

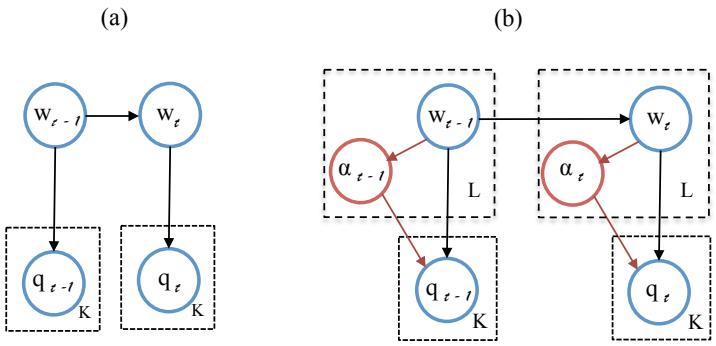

Figure 2: Graphical model comparison for (a) the conventional acoustic modeling and (b) posterior-based sparse modeling framework. The acoustic unit q_t denotes phones and w_t corresponds to words/language transcriptions. In (b) each w_t has an associated dictionary D_{w_t} and α_t is the sparse latent variable that identifies the mapping between acoustic observation q_t and words w_t based on D_{w_t}.

d_l has a sparse representation stated as

$$\underbrace{\begin{bmatrix} p(q_1|w_l) \\ p(q_2|w_l) \\ \vdots \\ p(q_K|w_l) \end{bmatrix}}_{d_l} = \underbrace{\begin{bmatrix} p(q_1|sw_1^{w_l}) & \cdots & p(q_1|sw_s^{w_l}) & \cdots & p(q_1|sw_{S_{w_l}}^{w_l}) \\ p(q_2|sw_1^{w_l}) & \cdots & p(q_2|sw_s^{w_l}) & \cdots & p(q_2|sw_{S_{w_l}}^{w_l}) \\ \vdots & & \vdots & & \\ p(q_K|sw_1^{w_l}) & \cdots & p(q_K|sw_s^{w_l}) & \cdots & p(q_K|sw_{S_{w_l}}^{w_l}) \end{bmatrix}}_{\text{Word manifold modeling dictionary:}\mathbf{D}_{w_l}} \times \begin{bmatrix} p(sw_1^{w_l}|w_l) \\ \vdots \\ p(sw_s^{w_l}|w_l) \\ \vdots \\ p(sw_{S_{w_l}}^{w_l}|w_l) \end{bmatrix} \tag{9}$$

where $sw_s^{w_l}$ denotes the sth sub-word unit of the word w_l, S_{w_l} represents the total number of (over-complete) "bases" to model the sub-space of word w_l.

Equations (8) and (9) lead us to a very intuitive and natural representation for continuous speech in terms of posterior features and word-to-subword hierarchical dictionaries. Thereby, the posterior-based sparse modeling dictionary is obtained as

$$\mathbf{D} = [\mathbf{D}_{w_1} \cdots \mathbf{D}_{w_l} \cdots \mathbf{D}_{w_L}] \tag{10}$$

The dictionary $\mathbf{D}$, has an internal partitioning defined by the boundaries of individual sub-dictionaries $\mathbf{D}_{w_l}$. Ideally, an input posterior feature z_t belonging to a realization of word w_l, when sparse coded using the dictionary above will have a sparse representation α_t such that only the atoms corresponding to the subdictionary $\mathbf{D}_{w_l}$, henceforth denoted as $\alpha_t^{w_l}$, will have non-zero values and $\alpha_t^{w_l}$ is expressed as

$$\begin{aligned} \alpha_t^{w_l} &= \left[p(sw_1^{w_l}|w_l) \ldots p(sw_s^{w_l}|w_l) \ldots p(sw_{S_{w_l}}^{w_l}|w_l)\right]^\top p(w_l|x_t) \\ &= \left[p(sw_1^{w_l}|x_t) \ldots p(sw_s^{w_l}|x_t) \ldots p(sw_{S_{w_l}}^{w_l}|x_t)\right]^\top \end{aligned} \tag{11}$$

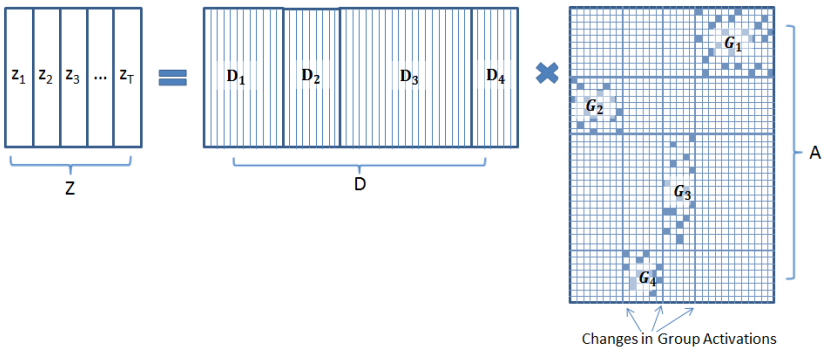

Figure 3: Given a sequence of acoustic features $\mathbf{Z}$, the sparse representation matrix $\mathbf{A}$ will have a sparse block structure associated to the word-specific dictionaries ($D'_{w_l}s$) where the inner block coefficients are sparse as well. This collaborative hierarchical sparsity structure is exploited in (Sprechmann et al., 2011) to devise an efficient C-HiLasso algorithm for the sparse recovery objective expressed in (5).

and

$$\alpha_t = [\alpha_t^{w_1} \dots \alpha_t^{w_l} \dots \alpha_t^{w_L}]^\top \tag{12}$$

When a sequence (matrix) of consecutive posterior feature vectors $\mathbf{Z} = [z_1, \dots, z_{t_1}, \dots, z_{t_2}, \dots, z_T]$, extracted from a speech utterance, is sparse coded using dictionary $\mathbf{D}$ (equation 8), it yields a sparse representation matrix $\mathbf{A} = [\alpha_1, \dots, \alpha_{t_1}, \dots, \alpha_{t_2}, \dots, \alpha_T]$ that exhibits a *collaborative hierarchical group sparsity structure* underlying its components. Consecutive posterior feature vectors $(z_{t_1}, \dots, z_{t_2})$ that belong to occurrence of the same word w_l excite only those atoms of dictionary $\mathbf{D}$ that correspond to the word manifold dictionary $\mathbf{D}_{w_l}$. Thus, they *collaborate* (in time dimension) to activate a higher level group $\alpha_t^{w_l}$ (as in equation 12) corresponding to $\mathbf{D}_{w_l}$. Moreover, the sparse representation α_t is sparse at two *hierarchical* levels: (i) in terms of the number of groups $\alpha_t^{w_l}$ activated (which is equal to one when only one word is spoken at a given time) and (ii) in terms of the non-zero coefficients of $\alpha_t^{w_l}$. This collaborative hierarchical structure is leveraged to devise the C-HiLasso algorithm for the objective function formulated in (5) (Sprechmann et al., 2011) and depicted in Figure 3. It may be noted that C-HiLasso forces activation of the same group(or groups) for all the posterior feature vectors that are being sparse coded together. Thus, an utterance with a sequence of words spoken one after another has to be sparse coded using C-HiLasso in a sliding window fashion. This ensures activation of a single group (word) in each position of the sliding window (more details in Sections 4.2 and 5.3).

In the following Section, we describe an application of the posterior-based sparse modeling formalism for automatic speech recognition task.

4. Posterior-based Sparse Modeling for Speech Recognition

The sparse representation, α_t's, can be directly used as posterior features in KL-HMM framework for speech recognition (Bahaadini et al., 2014) to improve the performance. In the present study, we focus on novel speech recognition paradigm that can be devised based on sparse modeling of posterior features.

Given a posterior feature z_t and the dictionary $\mathbf{D}$ defined in (10), we first obtain the sparse representation α_t using sparse recovery methods described in Section 2.2.2.[2] Defining $\alpha_t^{w_l}$ expressed in (11) as elements of α_t corresponding to the word-specific dictionary D_{w_l}, the posterior probability $p(w_l|x_t)$ for word w_l is estimated as

$$p(w_l|x_t) := \|\alpha_t^{w_l}\|_1 \tag{13}$$

assuming a union of disjoint events due to sparse recovery over the overcomplete dictionaries.

Consider a sequence of posterior features $\mathbf{Z}$ (estimated from acoustic features $\mathbf{X}$). A sequence of word posterior sparse representations $\mathbf{A}$ is obtained using the sparse recovery algorithms on $\mathbf{Z}$. Using the frame level word-posterior probabilities $p(w_l|x_t)$'s from equation (13), the maximum-a-posteriori word recognition can be obtained for $\mathbf{X}$ through

$$w_{\text{recognized}} := \arg\max_{w_l} p(w_l|\mathbf{X}) = \arg\max_{w_l} \prod_{t=1}^{T} p(w_l|x_t) \tag{14}$$

where T indicates the length of the test utterance in isolated word recognition.

4.1. Isolated Word Recognition

Although the conventional methods often exploit the full dictionary $\mathbf{D}$ for sparse reconstruction (Gemmeke et al., 2011; Bahaadini et al., 2014), sparse recovery for isolated word recognition tasks can also be done using word-specific dictionaries ($\mathbf{D}_{w_l}$) exploiting the prior knowledge on the dictionary partitioning. In this case, we obtain $\alpha_t^{w_l}$'s for each $\mathbf{D}_{w_l}$ directly, instead of α_t. This approach leads to word-wise sparse recovery with a caveat that the word posterior probabilities stated in (8) as α_t can not be directly obtained. The reason is that for each word w_l, a sparse representation $\alpha_t^{w_l}$ is computed through an independent non-competing sparse coding process using dictionary $\mathbf{D}_{w_l}$.

Instead, word recognition decisions for a sequence of posterior features $\mathbf{Z}$ can now be made using minimization of least-square reconstruction error over all dictionaries $\mathbf{D}_{w_l}$. The reconstruction error has been successfully applied for classification task (Wright et al., 2009) and linear predictive HMM (Kenny et al., 1990). If the reconstruction error for sparse recovery of z_t using dictionary $\mathbf{D}_{w_l}$

[2] To obtain the non-negative sparse word posterior probabilities, the Lasso algorithm is revised to project the coefficients at each iteration onto the non-negative orthant. These are separable constraints on the coordinates so it does not compromise the convergence of the method. Lastly, they are ℓ_1 normalized.

is denoted by

$$e_t^{w_l} = [e_t(1) \dots e_t(l) \dots e_t(L)]^\top, \text{where} \quad e_t(l) = \|z_t - \mathbf{D}_{w_l}\alpha_t\|_2^2$$

then word recognition for the complete sequence $\mathbf{Z}$ can be done using (14) and following *accumulation of errors* rule due to Gaussian noise model:

$$w_{\text{recognized}} := \arg\min_{w_l} \sum_{t=1}^{T} e_t^{w_l} \tag{15}$$

4.2. Continuous Speech Recognition

The difficulty in continuous speech recognition is rooted in the unknown word boundaries. Hence, T frames may encapsulate several classes with pauses in between. We learn a specific dictionary for sparse representation of the class of pause/silence. The pause state is already defined in the output layer of the neural network (c.f. Section 2.1) which makes it straightforward to distinguish the pause acoustic features from the training data. The neural network is not perfect in pause detection and learning a pause dictionary is beneficial for sparse modeling of continuous speech.

For continuous speech recognition, we can either employ sliding window based analysis or the C-HiLasso approach discussed earlier. We discuss these two approaches here.

4.2.1. Block-wise Search

Similar to isolated word recognition, sparse recovery can be done using word-specific dictionaries D_{w_l}'s. We just need to convert the reconstruction errors $e_t(l)$ into *empirical* word posterior probabilities. Let M denote the maximum value of $e_t^{w_l}$. The empirical word posterior probabilities are then obtained through

$$p(w_l|x_t) := \frac{M - e_t^{w_l}}{\|e_t\|_1} \tag{16}$$

The empirical probabilities in (16) can be used in a Viterbi decoder in a similar manner as the probabilities from equation (13).

4.2.2. C-HiLasso

Given test utterance $\mathbf{Z}_{\text{test}}$, a sliding window of appropriate length T' can be used to process a collection of frames $\mathbf{Z}_{\text{test}}^{t \dots t+T'-1}$ using C-HiLasso. The window length T' should be short enough to separate a single word and long enough to group the sequence of frames into a single consistent class. Hence, the choice of T' is not trivial and should be learned during the recognition task. It may be noted that the collaborative hierarchical Lasso requires the full dictionary $\mathbf{D}$ for computing sparse representation α_t. The word posterior probabilities from equation (13) are then simply employed to obtain the word sequence using a Viterbi decoder.

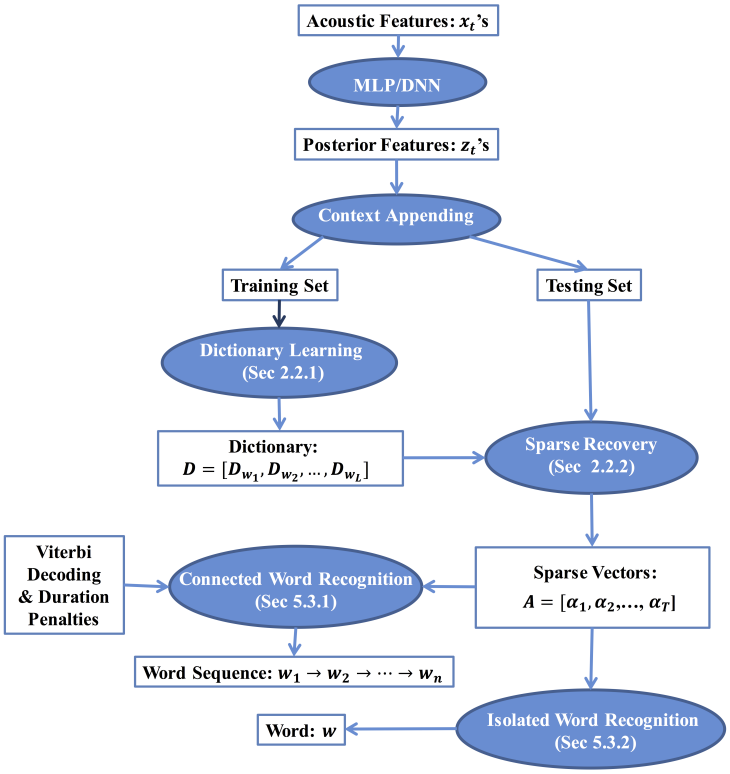

Figure 4: Flowchart of speech recognition system using the proposed posterior-based sparse modeling approach.

4.3. Modeling Temporal Information of Speech

The mechanism to account for temporal continuity or sequencing information of the acoustic features in our posterior-based sparse modeling framework is three-fold:

- Context Appending: A sequence of input frame-level posterior features are appended to form a segmental posterior feature. In other words, a context of c frames is concatenated in the form of $\tilde{z}_t = [z_{t-c}^\top \ldots z_t^\top \ldots z_{t+c}^\top]^\top$ as the input acoustic feature which is used for dictionary learning and word posterior sparse reconstruction[3]. This mechanism is referred to as *context appending* which is a typical approach to incorporate the dynamics of the

[3] As the input of the neural network is already context appended spectral features (see Figure 1), this approach is generally capable of exploiting a larger context than the conventional methods for word probability estimation.

	HMM	Template-matching	Sparse Modeling
Theory	Data is generated from probability distribution	Data live in space spanned by all training templates	Data live in a low-dimensional union of subspaces
Modeling	Gaussian/Multinomial fitting	Collection of Templates	Dictionary Learning
Algorithm	Viterbi Decoding	DTW Matching	Sparse Recovery

Table 1: Comparing HMM and DTW template matching and sparse modeling approaches to speech recognition.

features (Gemmeke et al., 2011; Bahaadini et al., 2014). Context appending incorporates time-dimension in the process of learning a dictionary.

- Structured Sparsity: The collaborative hierarchical Lasso stated in (5) exploits the hierarchical group sparsity structure to activate a single group/word in a collaborative fashion for a sequence of input posterior frames. This mechanism, referred to as *structured sparse modeling*, has not been considered in the earlier literature on exemplar-based sparse representation.

- Viterbi Decoding: When the dictionaries are learned for smaller speech units like phones/states, Viterbi algorithm can be used for decoding a word sequence from the phone posterior lattice given by the dictionary based sparse coding. Viterbi algorithm enforces the phone/state based language model underlying the word sequences.

Figure 4 illustrates the flow chart of the proposed posterior-based sparse modeling approach for speech recognition.

4.4. Relation with HMM and DTW

In this section, we discuss the links between sparse modeling, HMM and DTW sequence matching (acoustic modeling). Table 1 summarizes the key features of each approach. A good comparison between template based approaches and HMM is also given in De Wachter et al. (2007).

The HMM and DTW are devised to find the best match between the acoustic input and a set of reference exemplars. In case of HMM, the training exemplars are exploited to learn the hyper-parameters of a statistical model. Assuming that a probability distribution is a good hypothesis for the underlying generative process of the data, the HMM framework enables modeling the word manifold with a Markovian structure through the design of a parametric dictionary where each atom characterizes the underlying probability distribution. The parametric design approach can lead to better generalization of the model with fewer amount of training data. On the other hand, DTW is a non-parametric approach where the word manifold is assumed to be spanned by the exemplars

from the training data and the test acoustic input is characterized by the closest training exemplar. In that sense, the dictionary is the set of all training exemplars.

The sparse modeling approach relies on modeling the low-dimensional word manifold through *dictionary learning* rather than parametric design developed through HMM. In essence, the underlying process is assumed to generate the low-dimensional union of sub-spaces learned from the training data instead of being a Gaussian or multinomial distribution[4]. We will see in Section 5.3.1 that this new modeling paradigm can lead to better characterization of the MLP/DNN posterior features, while it bears the potential to be integrated with the design strategy of HMM framework. It is demonstrated that the k-sparse linear combination yields smaller characterization error for posterior feature vectors compared to the 1-sparse (DTW) and averaging (HMM with multinomial emission distribution) counterparts.

5. Experimental Analysis

We perform a series of experiments for empirical evaluation of the proposed approach. The experiments are devised to provide thorough analysis of the key features of this work and empirical insights into structured sparsity and contextual modeling. In addition, different computational methods to dictionary learning and sparse recovery are evaluated. Furthermore, we study the performance of the proposed posterior-based sparse modeling approach for *exemplar-based automatic speech recognition*. This task is studied in the context of isolated word recognition as well as continuous speech recognition and the performance is compared with the previous exemplar-based approaches.

5.1. Databases

Two databases are used: (1) Phonebook speech corpus (Pitrelli et al., 1995) recorded on single microphone channel at 16KHz, for isolated word recognition task and (2) Numbers database, a subset of Numbers 95 (Cole et al., 1995), recorded over telephone channel at 8KHz, for connected word recognition task. We perform two sets of experiments with Phonebook for isolated word recognition task - an easier 75 words vocabulary task and a more challenging 600 words vocabulary task. Each word has around 11 utterances, out of which we use 4 for learning dictionaries and the rest for testing. This setup is similar to the experiments in (Soldo et al., 2011).

For connected word recognition, we work on Numbers database, which has been created by picking, from Numbers 95 database, only those utterances that involve the 10 digits (*zero* to *nine*) and *oh* (alternative pronunciation for *zero*). Overall there are around 55k utterances, out of which we use 60% for training, 20% as development set and the rest for testing. Since the amount of training

[4]The multinomial distribution is considered in derivation of KL-HMM (Aradilla et al., 2008) which has been shown to be a suitable acoustic modeling framework for posterior features.

data is small for Phonebook, each dictionary is initialized with one of the four templates in the training data, and the rest are used for dictionary learning. Hence, the dictionary size is 25% of the size of training collection. For Numbers, we use a concatenation of 100 utterances for initializing the word-specific dictionaries, and the rest of training data for learning the dictionaries. As there are ∼3000 training exemplars per word, the dictionary size is ∼3% of the size of training collection. For both databases, the features used are similar to (Aradilla et al., 2008) as described in section 2.1.

5.2. Posterior-based Dictionary Learning and Sparse Recovery

In this section, we study different aspects of dictionary learning for sparse representation of posterior features.

5.2.1. Structured Sparsity

The high dimensional sparse representations exhibit some structures that can be exploited for speech recognition.

Sequencing pattern

We demonstrate that controlled initialization of the dictionary enables preserving the temporal information during the learning procedure. Using Phonebook data, the word-specific dictionary is initialized with an exemplar of the word. Dictionary learning explained in Section 2.2.1 leads to the atoms being updated such that the temporal evolution of the word is embedded in the sequence of the atoms. We can verify this hypothesis from the sparse representation of a sequence of acoustic features $\mathbf{Z}$ using the word-specific dictionaries. Figure 5 illustrates the sparse representation obtained for the sequence of the acoustic features of the word *'Accumulation'*. We can see that the sequencing pattern is exhibited when the correct dictionary, i.e., $\mathbf{D}_{\text{Accumulation}}$ is used for sparse recovery. On the other hand, the sequencing pattern is distorted when the wrong dictionary, e.g. $\mathbf{D}_{\text{Alleviatory}}$ is exploited.

The sequencing pattern can be justified from (9): Each of the dictionary columns behaves like the subword probabilities, $p(q_k|sw_s^{w_l})$ which are evolving with time. As a sequence of sw_s comprise the word w_l, the high dimensional sparse subword representations corresponding to $p(sw_s|w_l)$ exhibits the sequencing pattern. The sequencing pattern encourages us to look for mechanisms of incorporating the temporal information. One approach is through the use of structured sparse recovery based on C-HiLasso (5) that is studied hereafter.

Collaborative Hierarchical Sparsity

The collaborative hierarchical sparsity is explained in Section 3 for posterior-based sparse modeling. We can verify this intuition using C-HiLasso objective in (5) to obtain the word posterior probabilities of connected digits. Figure 6 demonstrates the sparse representation of a test digit sequence *0-2-1-4-4* using C-HiLasso when it is sparse coded using the complete dictionary $\mathbf{D}$. The results are contrasted with Figure 7 where the collaborative hierarchical sparsity structure is ignored for sparse representation.

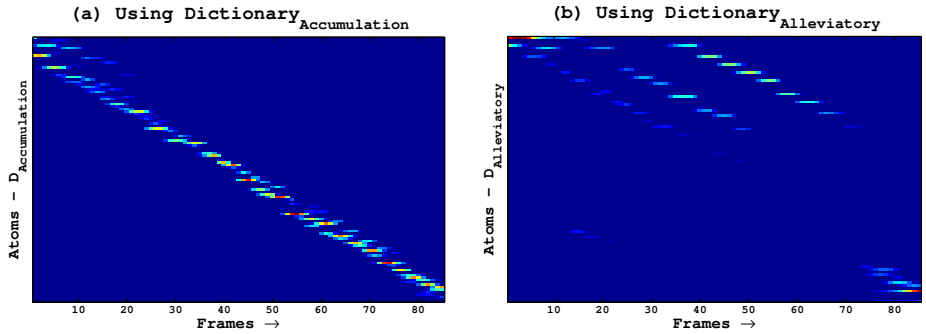

Figure 5: Sparse representation of the word "Accumulation" when the dictionary corresponds to (a) $\mathbf{D}_{\text{Accumulation}}$ and (b) $\mathbf{D}_{\text{Alleviatory}}$. The sequencing pattern is exhibited for the correct word hypothesis.

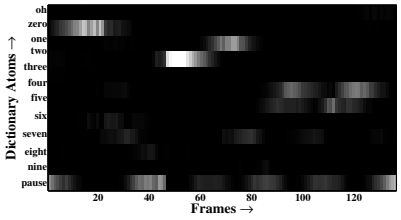

Figure 6: Sparse representation of connected digit sequence *0-2-1-4-4* using full dictionary ($\mathbf{D}$) and C-HiLasso sparse recovery.

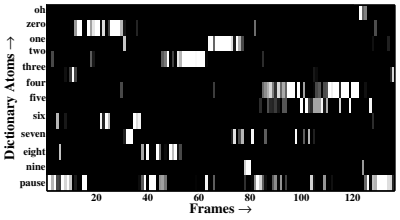

Figure 7: Sparse representation of connected digit sequence *0-2-1-4-4* using full dictionary ($\mathbf{D}$) and Lasso sparse recovery.

We can see that exploiting the structure sparsity of the sparse coefficients leads to better discrimination of the individual classes. The frame-level posterior features are used for this experiment. An alternative strategy to exploit the temporal information is devising the context-appended features.

5.2.2. Context Size Optimization

To incorporate the contextual information associated with temporal evolution of the features, one effective way is to append the posterior features of each frame with its neighboring posteriors. More specifically, for a context size of c, a frame-level posterior feature $z_t \in \mathbb{R}^K$ is mapped to segmental feature $\tilde{z}_t \in \mathbb{R}^{K(2c+1)}$ by appending c features on its right and left accordingly. This technique was successfully applied in (Bahaadini et al., 2014). Learning a dictionary this way improves the effectiveness of word-specific sub-dictionaries significantly as we will see below.

Figure 8 and 9 illustrate the improvement in isolated word recognition rate

for different context sizes using Phonebook and Numbers database respectively. A context size of $c = 20$ frames was found to be optimal for Phonebook corpus and the performance drops for larger c. This context size is applied for the rest of the evaluation on Phonebook data. On the other hand, we observe a consistent improvement in performance on Numbers recognition with increasing the context size. The difference can be justified due to the lack of training data in Phonebook.

The average word length of the Numbers corpus is $\sim$ 30 frames. Hence, longer contexts indicates that each feature vector represents the whole word. Hence, the sparse representation models the acoustic features as a linear combination of the full word exemplars. This concept has been applied successfully in (Gemmeke et al., 2011). However, (Gemmeke et al., 2011) use the full dictionary $\mathbf{D}$ for sparse recovery. We can verify that when standard Lasso is used, the word-specific sparse recovery, using $\mathbf{D}_{w_l}$'s, yields better word recognition performance than using the complete dictionary $\mathbf{D}$. The comparison of these two approaches is illustrated in Figure 8.

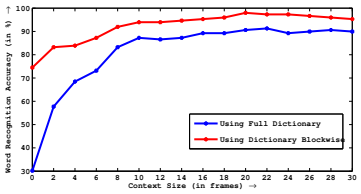

Figure 8: Optimization of context size for Phonebook word recognition using full dictionary (Gemmeke et al., 2011; Bahaadini et al., 2014) and word-specific dictionaries for sparse recovery. The best performance is achieved at context of 20 frames using word-specific dictionaries.

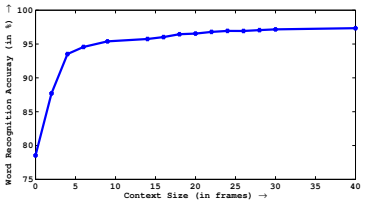

Figure 9: Consistent word recognition improvement with the increase in context size for Numbers data. At the context size above 30 frames, the atoms correspond to full-word exemplars which have been successfully applied for spectral-based sparse representation (Gemmeke et al., 2011).

It may be noted that the use of context appended features is complementary or even alternative to the collaborative hierarchical structured sparse recovery. In fact, our experiments on continuous speech recognition presented in Section 5.3.2 reveals that once the "optimal" context size is applied, the block-wise sparse recovery using word-specific dictionaries outperforms C-HiLasso. Nevertheless, the compromise for smaller context and structured sparse recovery is an interesting feature of this work.

5.2.3. Comparison of Dictionary Learning and Sparse Recovery Algorithms

Dictionary learning and sparse recovery are the two pillars of sparse modeling framework. We conduct some experiments using the state-of-the-art techniques to learn the dictionary of posterior-based exemplars and obtain the word posterior sparse representation for speech recognition. The evaluation is performed

on Phonebook 75-vocabulary isolated word recognition task. The results are listed in Table 2.

The best recognition performance is obtained using the online dictionary learning algorithm with Lasso sparse recovery (see (Mairal et al., 2010)) with an accuracy of 97.2%. The online dictionary learning algorithm has been found to work fast with LARS Lasso (Efron et al., 2004) with higher accuracies. K-SVD performs poorly in comparison. One of the weakness of K-SVD is that the algorithm can get stuck in local minima because of the non-convexity of the problem (Aharon et al., 2006).

	Lasso	OMP	GKL
Online Algorithm	**97.2**	93.5	76.5
KSVD	55.8	88.9	8.4

Table 2: Word recognition rate (%) on Phonebook 75-vocabulary dataset using different computational methods to dictionary learning and sparse recovery. In this study we consider the online dictionary learning (Mairal et al., 2010) and KSVD (Aharon et al., 2006) algorithms for learning the dictionary of segmental posterior exemplars ($c = 20$). The LARS implementation of LASSO (Efron et al., 2004), OMP (Tropp and Wright, 2010) and generalized Kullback-Leibler divergence (GKL) (6) sparse recovery (Gemmeke et al., 2011) algorithms are used for reconstruction of word posterior sparse representation. The word recognition is obtained through (14)-(16).

5.2.4. Dictionary Learning vs Collection of Exemplars

Finally, we can verify the hypothesis that dictionary learning performs better than the use of all training exemplars for sparse representation. In isolated word recognition experiment on Phonebook 75-vocabulary dataset, a single exemplar is used as a warm start for dictionary initializing. The remaining 3 exemplars in the training set are then used for updating the dictionary columns using Algorithm 1. Alternatively, 4 training exemplars are concatenated to form a dictionary for sparse representation. A similar comparison was done for connected digit recognition on Numbers database, where we can either learn word-specific dictionaries or we can directly represent each word using all training exemplars (Gemmeke et al., 2011). The results are listed in Table 3. We can see that the dictionary learning procedure is quite effective; it can benefit from the abundance of the training data, while it enables us to keep the dimensionality of the exemplar space small and at the same time improve the performance.

5.3. Experiments on Automatic Speech Recognition

In this section, we focus on evaluation of the proposed system for automatic speech recognition.

5.3.1. Exemplar-based Isolated Word Recognition

The isolated word recognition evaluation is conducted on Phonebook database. The exemplar-based sparse modeling using spectral features (Gemmeke et al., 2011) yields less than 50% accuracy on 75-vocabulary recognition

Task	Dictionary Learning	Collection of Exemplars
Phonebook	**97.2**	97
Numbers	**85.4**	78.6

Table 3: Comparing the speech recognition accuracy (%) on Phonebook (isolated word recognition) and (connected word recognition - Section 5.3.2) using dictionary learning versus collection of exemplars. The size of training data in Phonebook is small. In this case the dimension of dictionary exemplars (number of learned atoms) is 25% of the full training set. The size of training data in Numbers corpus is large. In this case the dimension of dictionary exemplars (number of learned atoms) is $\sim$3% of the full training set.

database. This observation can be justified as the training data is really scarce and does not meet the requirement for the system developed by (Gemmeke et al., 2011).

The prior studies have shown that posterior features perform well when the training data is limited to a few exemplars using DTW template matching (Aradilla et al., 2009; Soldo et al., 2011). Hence, the DTW-based word recognition is our benchmark for this study. We compare the performance of our posterior-based sparse modeling framework with an equivalent DTW template matching system that uses Euclidean distance metric. Both systems are given the same training set of 4 utterances per word and the rest of the utterances for testing. While dictionary learning approach utilizes (14)-(16) for decision making, the DTW approach is based on finding the minimum distance template from the training sets of all words. Table 4 shows the results for these experiments on 75-vocabulary and 600-vocabulary sets. We can see that the proposed posterior based sparse modeling framework outperforms the similar DTW template-matching system. The online dictionary learning algorithm along with LARS Lasso sparse recovery (Mairal et al., 2010) are used for word posterior sparse reconstruction.

System	PB75	PB600
DTW	84.7	73.5
Sparse Modeling	**97.8**	**93.2**

Table 4: Isolated word recognition accuracies (in %) for Phonebook database on 75-vocabulary (PB75) and 600-vocabulary (PB600) sets.

The word recognition accuracy of the HMM/MLP system presented in (Pinto et al., 2009)[5] is 98.8% for 75-vocabulary set and 96.0% for 600-vocabulary set. Previous work by (Soldo et al., 2011) showed that the best results were ob-

[5]This system is task-independantly trained so the scenario is more difficult than the presented scenario due to unseen words.

tained by DTW template matching using weighted symmetric KL divergence and it outperforms the HMM/MLP system. Hence, our future work will consider devising the dictionary learning and sparse recovery algorithms tailored for this "optimal" distance measure.

Union of Subspaces Model

We recall our discussion in Section 4.4 on the theoretical assumption underlying different modeling strategies for speech recognition. Given a pool of training exemplars, sparse modeling, HMM and DTW take different approaches to model the word manifolds. Sparse modeling assumes a union of subspace model and learns an overcomplete "basis" set for sparse representation using the training data. The test exemplar is characterized as a k-sparse linear combination of the "basis" vectors. On the other hand, HMM assumes that data is generated through a specific probability distribution (e.g. Gaussian or multinomial) and learns the associated hyper-parameters. This approach can also be seen as designing a parametric dictionary for word manifold. DTW assumes that the whole space is spanned by the training exemplars in which the test exemplar is best represented by the closest training data. The EM procedure for learning the parameters of an HMM with multinomial emission likelihood indicates that the average of all training exemplars associated to each state can characterize the subspace of an HMM state (Aradilla et al., 2008). In contrast, the DTW approach assumes a 1-sparse characterization of the test exemplar and the sparse modeling approach employs a k-sparse linear combination of the training data.

Our hypothesis is that sparse modeling is more accurate in characterization of the training data. More specifically, representing a posterior exemplar as a k-sparse combination of the training exemplars is more accurate than 1-sparse (as in DTW) or averaging (as in KL-HMM) characterization. The accuracy in this context is quantified in terms of weighted symmetric KL divergence as it was shown to be an appropriate distance measure in posterior feature space (Aradilla et al., 2008).

To validate this hypothesis, we perform a simple experiment of template matching using DTW for 75-vocabulary set of Phonebook. Out of 11 utterances for each word, we keep 4 utterances as training templates and use the rest for testing. All possible averaging of the 4 exemplars, i.e. $\binom{4}{k,\ \forall k \in \{1,2,3,4\}}$ are obtained by DTW matching of the selected templates followed by averaging the corresponding elements to obtain a single template. We then quantify the distance of the test exemplars with the newly constructed templates. The smaller distance indicates better characterization of the test templates using the training data. This experiment is run for all test data. We observe that only 4.9% of the exemplars have the least characterization error using a single closest template (DTW assumption). Moreover, only 9.7% are best characterized by the model obtained from averaging the full training set (KL-HMM assumption (Aradilla et al., 2008)). On the other hand, all remaining 85.4% of the exemplars have the least characterization error using the templates which are obtained as a combination of a few (2 or 3) training exemplars. This observation

confirms the hypothesis of the effectiveness of the union of subspace approach to model the posterior feature space.

5.3.2. Exemplar-based Continuous Speech Recognition

The continuous speech recognition evaluation is conducted on Numbers database. As indicated earlier, the sparse representation, α_t's can be directly used as posterior features in KL-HMM framework for speech recognition (Bahaadini et al., 2014) and it has been shown to improve the performance. In this study, we devise a speech recognition system based on sparse modeling of posterior features. The prior works (Gemmeke et al., 2009, 2011) on exemplar-based sparse representation use spectral features for connected digit recognition task, whereas we work with MLP/DNN based posterior features. Moreover we use dictionaries in place of collection-of-exemplars to estimate word/phone likelihoods and use them to decode the most likely sequence of digits relying on Viterbi dynamic programming.

Word-specific Dictionary Learning

Previous approaches employ a collection of all exemplars for sparse representation. However, our preliminary evaluation (c.f. Figure 8) suggests the use of word-specific dictionaries for block-wise sparse recovery. We compare both approaches in this experiment. A sequence of 17 frames ($c = 8$) is concatenated to encode the dynamics of the features. This analysis window is shifted by one-frame at a time as it was shown to yield the best recognition results (Gemmeke et al., 2009). Furthermore, a sequence of $T' = 3$ such concatenated frames are considered for C-HiLasso to exploit the collaborative group sparsity structure underlying the sequence of sparse representations. Recall the discussions in Sections 4 and 5.2.1-5.2.2 that context appending and collaborative hierarchical sparsity are our mechanisms to incorporate the sequencing information underlying the acoustic features.

The Viterbi decoder as explained in Gemmeke et al. (2009) is implemented to decode the word sequence. For each digit, we learn the maximum and minimum durations from the training set. The Viterbi decoder applies duration penalties to all the paths where these duration constraints are violated. No language model is used for this task. The results are presented in Table 5 (systems 2–3). The word-based dictionary consists of ∼3000 exemplars. Posterior-based system using word dictionaries performs better (14.6% WER) as compared to the baseline exemplar-based approach using collection of exemplars (system 1). Furthermore, the performance is comparable to the best results of Gemmeke et al. (2009) on word-based labeling using a collection of 16000 spectral features and context size of 35 frames. This observation confirms the hypothesis that posterior features are more suitable than the spectral features for exemplar-based sparse modeling approach to continuous speech recognition. Moreover, dictionary learning is central to gain significant improvement in performance.

In addition, we can see that context appended segmental features are quite effective in exploiting the temporal information. Once again, the dictionary

learning outperform the conventional use of the collection of exemplars significantly. The window size for C-HiLasso is an important parameter optimizing which can lead to better recognition results. However, the C-HiLasso approach performs worse than the block-wise search. The reason can be associated to downsampling of the sparse recovery problem by a factor of 12 in the block-wise approach.

A major problem with the word-based labeling of the training exemplars is associated with the occurrences of the repeated words, e.g. 4-4-4. In such cases, distinguishing among the same classes is hard if there is no pause in between and the word durations are relatively short. To tackle this problem, we use phone-based labeling and decoding.

Phone-specific Dictionary Learning

For the training set, we generate the phone alignments for the digit sequences using 27 phones using a Viterbi decoder. Each digit is expressed by a sequence of 3 to 5 phones except 'oh' which constitutes of a single phone. Phone-based dictionaries are learned from the training set using these alignments. We label the same phone in two different digits with a different tag so as to learn two independent dictionaries for it from its occurrence in two different contexts. With this procedure, we learn a total of 36 phone dictionaries. A pause dictionary is used in this case as well. For phone-based evaluation, a sequence of 5 frames are concatenated ($c = 2$) and we use the similar sliding window mechanism for analysis as in the previous experiment. We generate phone probabilities for each input feature vector and pass them to a Viterbi decoder. Viterbi paths here are restricted to valid phone sequences using topology of the possible transitions. Penalties associated with word transitions and duration penalties for phones are also used. The results are presented in Table 5 (systems 4–5). We get a WER of 12.5% with phone-based dictionaries. Handling repeated digits becomes a trivial issue with phone-based dictionaries. The phone-specific dictionary is also applicable to model the unseen words.

#	System	WER(in %)
1	Collection of (posterior) exemplars	21.4
2	Word Dictionary (block-wise search)	14.6
3	Word Dictionary (C-HiLasso)	18.5
4	Phone Dictionary (block-wise search)	**12.5**
5	Phone Dictionary (C-HiLasso)	17.7

Table 5: Exemplar-based Connected Digit Recognition on Numbers database. Word Error Rate (WER) is obtained by Levenshtein distance.

Hybrid Exemplar-based/Probabilistic HMM Decoding

A major difference from the previous exemplar-based sparse representation approaches (Gemmeke et al., 2009, 2011) comes from the fact that they are hybridized with a conventional HMM. A state label matrix for each exemplar

(obtained from a conventional HMM system) is rescored using sparse recovery based likelihoods before Viterbi decoding. In contrast, system 2 to system 5 in Table 5 decode the obtained sparse recovery based likelihoods directly without using any HMM based state labels. A hybrid dictionary-based/HMM decoding system was also implemented exactly analogous to hybrid exemplar-based/HMM in (Gemmeke et al., 2011). In this experiment, we get state labels for each frame using a conventional GMM-HMM system (Povey et al., 2011). State labels assigned to the training data are used to learn the state-specific dictionaries. In total, there are 111 states including 3 states for pause. On the test data, these state-specific dictionaries generate sparse state likelihood scores which are hybridised with the HMM state labelings for Viterbi decoding thereafter. This hybrid system works at a WER of 10.0% and doesn't show improvement upon the GMM-HMM system (9.4%). This observation is in line with the results of (Gemmeke et al., 2011) in clean condition experiments. The HMM-MLP system (Aradilla, 2008) works at 7.2% WER.

6. Conclusions

The present work demonstrates a novel study of exemplar-based sparse modeling for speech recognition using neural network based posterior features. In this context, the posterior features outperform the conventional spectral features. In addition, we show that exemplar-based speech recognition systems can benefit from dictionary learning algorithms to reduce the dimension of the training exemplars into a small learned "basis" components for characterizing the low-dimensional manifolds associated to the linguistic units (e.g. words, phones). We confirm the hypothesis that the posterior features can be effectively characterized using a union of subspace model. The theory of exemplar-based sparse modeling is tightly related to the theory of statistical speech recognition.

We observe that the temporal sequencing information can be exploited by using either segmental features or collaborative hierarchical sparse recovery. The advantage of structured sparse reconstruction is that the sequencing Markovian structure underlying both DTW and HMM based systems can be replaced by a more relaxed structure in sparse modeling framework where the representation coefficients collaborate to activate a sparse set of linguistic units (e.g words or phones). The choice of an appropriate group size for structured sparsity is a parameter dependent on speech units being recognised. This framework can bring us different advantages in continuous speech recognition system by alleviating the requirement for identifying the inter-dependency of the acoustic features prior to recognition. Still we acknowledge that capturing temporal properties of speech using dictionary-based sparse representation approaches is an open issue for further investigation. Future work lies in integrating the proposed system with other components of a traditional automatic speech recognition system. Since we obtain the sparse representation as a posterior space, we can construct HMMs with states corresponding to the dictionary atoms where the emission probabilities are characterized as a union of subspaces model. This *sparse-HMM* formalism can integrate the qualifications of exemplar modeling and HMM to

benefit from the strengths of both systems, in particular the power of dictionary learning in better characterization of the posterior features compared to the alternative methods (Section 5.3.1-Union of Subspaces Model). Furthermore, alternative distance measures in posterior feature space such as weighted symmetric KL have been shown to better capture the relation of these acoustic features. Hence, our future plan will consider devising the dictionary learning and sparse recovery algorithms tailored for this "optimal" distance measure.

Previous work based on exemplar-based sparse representations (Gemmeke et al., 2011) has been shown to give promising results on ASR in noisy conditions especially at lower signal-to-noise ratio environments. This provides motivation for developing a noise robust ASR system using the proposed posterior features and dictionary learning based framework. Possible applications of such a system can also include other mismatched conditions like accented speech or multilingual ASR.

Acknowledgment

The research leading to these results has received funding from by SNSF project on "Parsimonious Hierarchical Automatic Speech Recognition (PHASER)" grant agreement number 200021-153507. The authors would like to acknowledge Dr. David Imseng for his assistance with speech recognition experiments.

We also would like to acknowledge the anonymous reviewers for the insightful comments and remarks to improve the quality and clarity of the manuscript.

References

Aharon, M., Elad, M., Bruckstein, A., 2006. KSVD: An algorithm for designing overcomplete dictionaries for sparse representation. Signal Processing, IEEE Transactions on 54, 4311–4322.

Aradilla, G., 2008. Acoustic models for posterior features in speech recognition. Ph.D. thesis. École Polytechnique Fédéral de Lausanne (EPFL).

Aradilla, G., Bourlard, H., Magimai-Doss, M., 2008. Using KL-based acoustic models in a large vocabulary recognition task., in: INTERSPEECH, pp. 928–931.

Aradilla, G., Bourlard, H., et al., 2009. Posterior features applied to speech recognition tasks with user-defined vocabulary, in: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, IEEE. pp. 3809–3812.

Asaei, A., Bourlard, H., Cevher, V., 2011. Model-based compressive sensing for multi-party distant speech recognition, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Asaei, A., Picart, B., Bourlard, H., 2010. Analysis of phone posterior feature space exploiting class-specific sparsity and MLP-based similarity measure, in: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP).

Bahaadini, S., Asaei, A., Imseng, D., Bourlard, H., 2014. Posterior-based sparse representation for automatic speech recognition, in: Proceeding of Interspeech.

Cole, R.A., Noel, M., Lander, T., Durham, T., 1995. New telephone speech corpora at cslu.

Davis, G., Mallat, S., Avellaneda, M., 1997. Adaptive greedy approximations. Constructive approximation 13, 57–98.

De Wachter, M., Matton, M., Demuynck, K., Wambacq, P., Cools, R., Van Compernolle, D., 2007. Template-based continuous speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on 15, 1377–1390.

Devijver, P.A., Kittler, J., 1982. Pattern recognition: A statistical approach. volume 761. Prentice-Hall London.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al., 2004. Least angle regression. The Annals of statistics 32, 407–499.

Gemmeke, J., Ten Bosch, L., Boves, L., Cranen, B., 2009. Using sparse representations for exemplar based continuous digit recognition, in: Proc. EUSIPCO, Citeseer. pp. 24–28.

Gemmeke, J.F., Virtanen, T., Hurmalainen, A., 2011. Exemplar-based sparse representations for noise robust automatic speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on 19, 2067–2080.

Jansen, A., Niyogi, P., 2006. Intrinsic fourier analysis on the manifold of speech sounds, in: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.

Kenny, P., Lennig, M., Mermelstein, P., 1990. A linear predictive hmm for vector valued observation with application to speech recognition. IEEE Transactions on Acoustic Speech and Signal Processing 38.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., 2010. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research (JMLR) 11, 19–60.

Pinto, J.P., Magimai.-Doss, M., Bourlard, H., 2009. MLP based hierarchical system for task adaptation in asr, in: Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU).

Pitrelli, J., Fong, C., Wong, S., Spitz, J., Leung, H., 1995. Phonebook: a phonetically-rich isolated-word telephone-speech database, in: Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, pp. 101–104 vol.1. doi:10.1109/ICASSP.1995.479283.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., et al., 2011. The kaldi speech recognition toolkit .

Sainath, T.N., Ramabhadran, B., Nahamoo, D., Kanevsky, D., Sethy, A., 2010. Sparse representation features for speech recognition., in: INTERSPEECH, pp. 2254–2257.

Sainath, T.N., Ramabhadran, B., Nahamoo, D., Kanevsky, D., Van Compernolle, D., Demuynck, K., Gemmeke, J.F., Bellegarda, J.R., Sundaram, S., 2012. Exemplar-based processing for speech recognition: An overview. Signal Processing Magazine, IEEE 29, 98–113.

Sainath, T.N., Ramabhadran, B., Picheny, M., Nahamoo, D., Kanevsky, D., 2011. Exemplar-based sparse representation features: From TIMIT to LVCSR. Audio, Speech, and Language Processing, IEEE Transactions on 19, 2598–2613.

Soldo, S., Magimai-Doss, M., Pinto, J., Bourlard, H., 2011. Posterior features for template-based ASR, in: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, IEEE. pp. 4864–4867.

Sprechmann, P., Ramirez, I., Sapiro, G., Eldar, Y.C., 2011. C-hilasso: A collaborative hierarchical sparse modeling framework. Signal Processing, IEEE Transactions on 59, 4183–4198.

Stevens, K.N., 1998. Acoustic phonetics .

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) , 267–288.

Tropp, J.A., Wright, S.J., 2010. Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE 98, 948–958.

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y., 2009. Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31, 210–227.