BROADCAST NEWS PHONEME RECOGNITION BY SPARSE CODING

Joseph Razik[1], Sébastien Paris[2] and Hervé Glotin[1]

[1] *LSIS/DYNI, Université du Sud Toulon-Var, Avenue de l'Université - BP20132, 83957 LA GARDE CEDEX - FRANCE.*

[2] *LSIS/DYNI, Université Aix-Marseille, Domaine universitaire de Saint Jérôme, Avenue Escadrille Normandie Niemen, 13397 MARSEILLE Cedex 20, FRANCE.*

{*razik, glotin*}*@univ-tln.fr, sebastien.paris@lsis.org*

Keywords: MFCC, GMM, Sparse Coding, Large-Scale SVM, Explicit Feature Maps.

Abstract: We present in this paper a novel approach for the phoneme recognition task that we want to extend to an automatic speech recognition system (ASR). Usual ASR systems are based on a GMM-HMM combination that represents a fully generative approach. Current discriminative methods are not tractable in large scale data set case, especially with non-linear kernel. In our system, we introduce a new scheme using jointly sparse coding and an approximation additive kernel for fast SVM training for phoneme recognition. Thus, on a broadcast news corpus, our system outperforms the use of GMMs by around 2.5% and is computationally linear to the number of samples.

1 INTRODUCTION

In recent years major advancements have been achieved in speech processing. However, robust speech recognition still remains a challenging task. Current systems are still exhibiting difficulties when dealing with real-life conditions such as: multiple-speakers without training, noise, background music, spontaneous speech. The most common architecture of an automatic speech recognition (ASR) system is based on a generative framework and more specially a GMM-HMM approach (Gaussian Mixture Model - Hidden Markov Model). Prior to fully decoding sentences, a kind of phoneme recognition module is the first and most crucial part of the ASR system.

The latter is usually modeled by a GMM with a given number of components, large enough to capture the intra-variability of the phonemes (Huang et al., 2001). It assumes that the conditional pdf to each phoneme's class has a parametric form and that is comprised a mixture of normal pdfs. In practice, this strong assumption is mainly verified when training data is sufficiently available, but can introduce some over-fitting for less populated classes.

Recently, in vision systems, discriminative approaches combining bag-of-features and large-scale classifiers have shown dramatic improvement versus generative methods (Yang et al., 2009). These new approaches relie on three basic ingredients: (i) an unsupervised data encoding, (ii) feature extraction from a learned dictionary with pooling, and (iii) fast SVM (Support Vector Machines) for classification.

For the first step, sparse learning (see in (Lin et al., 2008; Hsieh et al., 2008; Wang et al., 2010; Smit and Barnard, 2009; Sivaram et al., 2010; Mairal et al., 2010)) allows a smaller re-construction error with few basis vectors, involving discriminative vectors' dictionary properties. After applying sparse code pooling, we obtain features descriptors with only positive or null values. In this case, specialized kernels such as intersection histogram kernel, offer the state of the art classification performances (Maji et al., 2009). In (Vedaldi and Zisserman, 2011), such kernels can be efficiently approximated via the feature map framework, involving fast training (linear in number of training samples (Fan et al., 2008)).

In this paper, we propose a new phoneme recognition system based on MFCC (Mel Frequency Cepstral Coefficient) sparse coding and fast non-linear training.

First, in section 2 we will describe the audio MFCC parameters as our input features, then in section 3 we present a short overview of the GMM. Section 4 provides an introduction of the sparse coding framework. Pooling methods are presented in section 5. The large-scale linear SVM is reviewed in section 6 with also the feature maps homogeneous additive kernel approximation method. Finally, sections 7 and

8 are dedicated to corpus presentation and results.

2 PARAMETERS

The parameters extracted from the audio signal are based on MFCC (Mel Frequency Cepstral Coefficients) (Davis and Mermelstein, 1980; Rabiner and Juang, 1993) with CMS (Cepstral Mean Subtraction) normalization. The frame-shift for their computation is 10 ms.

Additionally to the static MFCC coefficients ($C_0 \ldots C_{12}$), we compute also dynamic information values as their first derivative and variance. More precisely, we compute and concatenate to the final parameter vector the variance, and several approximated derivative coefficients according to the ranges between points for this calculation (20, 62, 125 and 250 ms). The usual approximation at time t is done by regression with coefficients at $t-\alpha$ and $t+\alpha$ (Young et al., 1995). Moreover, two size of the analysis window is used for the computation of the MFCC (16 and 32 ms). Thus, the final parameter vector is the concatenation of each kind of parametrization and its total dimension is 260, 13 statics and 13 dynamics by 5 kinds of dynamic coefficients and 2 window sizes.

2.1 Whitening

Several pre-processing methods of the parameter vectors could be applied in order to have the data better conditioned. The first simple pre-processing step is to center the data, i.e. to make the mean of the parameter vector equals zero. In fact, the static MFCC coefficients are already computed as centered with the CMS normalization but the dynamic coefficients are not.

In a second step, the entire data set of the vectors may be whitened. This process is commonly used in deep learning domain but not so frequently in speech recognition (Ranzato et al., 2010). The whitening process consists in decorrelating the data and making their variances equal to unity by the use of an eigen-value decomposition (EVD) (Hyvärinen and Oja, 2000).

More precisely, if we called X the set of parameter vectors, X is then linearly transformed into a whiten set $\tilde{X}$ with the property that $E\{\tilde{X}\tilde{X}^T\} = \mathbf{Id}$, the identity matrix. By using an EVD decomposition of the preceding covariance matrix, we obtain the relation $E\{\tilde{X}\tilde{X}^T\} = E\Delta E^T$, where E is the orthogonal matrix of eigenvectors of $E\{\tilde{X}\tilde{X}^T\}$ and Δ is the diagonal matrix of its eigenvalues: $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_n)$. The estimation of the whiten vectors $\tilde{x}$ can now be obtained by the following equation:

$$\tilde{x} = E\Delta^{-1/2}E^T x \tag{1}$$

where $\Delta^{-1/2} = \mathrm{diag}(\delta_1^{-1/2}, \ldots, \delta_n^{-1/2})$.

Thus, besides the raw parameter vectors we assess the use of a whitening process on our MFCC data and the impact on the recognition rate.

3 BASELINE SYSTEM: GMM

We compare our phoneme recognition method to one based on Gaussian Mixture Models (GMM) that is considered as a reference system in phoneme recognition task.

GMMs are used as a generative classifier modeling data classes as a mixture of M Gaussian pdfs and expressed as:

$$G(\mathbf{x}|y;\theta) = \sum_{i=1}^{M} w_i \mathcal{N}(x, \mu_i, \Sigma_i), \qquad \sum_{i=1}^{M} w_i = 1 \tag{2}$$

for which, with vectors of n dimensions, the continuous pdf is defined as:

$$\mathcal{N}(\mathbf{x}, \mu, \Sigma) \triangleq \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{3}$$

The w_i coefficients represent the weights of each Gaussian pdf composing the conditional pdf $G(\mathbf{x}|y;\theta)$. For each class $y = j$, $j = 1, \ldots, V$, the training process consists in learning parameters $\theta_j \triangleq \{w_{j,i}, \mu_{j,i}, \Sigma_{j,i}\}$, $i = 1, \ldots, M$ by an EM algorithm (Huang et al., 2001) and particularly with the HTK software (Young et al., 1995). Moreover, to avoid over-fitting we assume during the GMMs training that the covariances matrices $\Sigma_{j,i}$ are diagonals and we add an extra regularization term. For all the $V = 41$ classes, we will learn the conditional pdf $G(\mathbf{x}|y;\theta)$ by varying the number of components that maximizes the accuracy, from $M = 1$ to $M = 256$.

4 FROM VECTOR QUANTIFICATION TO SPARSE CODING

Super-vectors coupled with a supervised training algorithm represents a promising discriminating approach (Arous and Ellouze, 2003; Tomi Kinnunen, 2010). We propose to build such super-vectors using sparse coding and a pooling procedure. Let X

be the set of $n \times N$-dimensional MFCC matrix extracted from the audio, *i.e.* $X \triangleq [x_1, \ldots, x_N] \in \mathbb{R}^{n \times N}$. Let D be a dictionary made of K vectors such $D \triangleq [d_1, \ldots, d_K] \in \mathbb{R}^{n \times K}$ and trained from data. This dictionary clusters/resumes all data X into K codebook vectors.

From this trained D, in the traditional vector quantization (VQ) approach, each vector x_i of X is assigned to only one d_j such that:

$$d_j = \underset{k=1,\ldots,K}{\arg\min} \|x_i - d_k\|_2 \tag{4}$$

Let be $C = [c_1, \ldots, c_K] \in \mathbb{R}^{n \times K}$ the VQ matrix for which each c_i vector has only one component $c_i^j \neq 0$ (corresponding to the d_j codebook vector from the preceding equation (4). The associated VQ optimization problem is formulated as follows:

$$\underset{D,C}{\arg\min} \sum_{i=1}^{N} \|x_i - Dc_i\|_2^2 \quad s.t. \quad \|c_i\|_{\ell_0} = 1, \forall i \tag{5}$$

where $\|x\|_{\ell_0}$ designs the pseudo zero-norm, *i.e.* only one element of x is equal to 1, others are set to 0. In equation (5), D,C must be optimized jointly by the Kmeans algorithm or variants for example.

In the sparse coding (SC) approach, the difference is that each vector x_i can be expressed as a linear combination of the vectors of the dictionary D and not only by one of it. Then, the problem to solve is extended as in the following equation:

$$\underset{D,C}{\arg\min} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_{\ell_1} \quad s.t. \quad \|c_i\|_{\ell_1} = 1 \tag{6}$$

The regularization term λ coupled with the ℓ_1 norm, as seen in the equation (6), ensures the sparsity of the optimized codes. Unfortunately, this joint constrained optimization does not have a convex explicit formulation and the resolution is done in two steps, repeated iteratively until convergence.

The first step consists in updating the current estimation of the dictionary $\widehat{D}_{t+1}$ given current sparse codes $\widehat{C}_t$ via a block coordinate descent optimizer. The second step consists in updating sparse codes $\widehat{C}_{t+1}$ given the current dictionary $\widehat{D}_{t+1}$ via a LASSO algorithm (Mairal et al., 2009).

5 SPARSE CODES POOLING TO CONSTRUCT A NEW AUDIO DESCRIPTOR

Each phoneme realization p_i, $i = 1, \ldots, P$, where P defines the total number of phonemes in the data set, is associated with F_i MFCC vectors x_l, $l = 1, \ldots, F_i$ (of dimension $n = 260$ and such that $\sum_{i=1}^{P} F_i = N$, the total number of audio parameter vectors). In order to construct the new proposed audio feature vector $z_i = [z_1, \ldots, z_K]$ for this phoneme, all associated sparse codes $\{c_l\}$, $l = 1, \ldots, F_i$ are projected/pooled by one of the two following methods:

- Mean pooling:

$$z_i^j = \frac{1}{F_i} \sum_{l=1}^{F_i} |c_l^j| \tag{7}$$

- Max pooling:

$$z_i^j = \max(|c_1^j|, \ldots, |c_{F_i}^j|) \tag{8}$$

The set of the P features $\{z_i\}$ and their corresponding label will be trained efficiently with a large-scale linear SVM. Since sparse dictionary learning coupled with max-pooling method produce almost perfectly linear separable descriptors, a fast linear SVM solver such as Liblinear is preferred. This approach can be perceived discriminative since we don't focus on the data densities to classify, but only on the frontiers of separation.

6 LARGE-SCALE LINEAR SUPPORT-VECTOR MACHINE

Once the descriptors are computed, a large-scale linear SVM can be used as classifier. For training the model efficiently, according to the Vapnik theory (Vapnik, 1998), we aim to minimize the structural error in order to generalize performances on unseen data. This leads to find a binary classifier separating classes based on the maximum margin principle. Several algorithms exist that aim to find models maximizing such margins, for example neural networks with Generalized Relevance Learning Vector Quantization (GLRVQ) (Hammer et al., 2004), some variant of adaboosting (Rudin et al., 2007) and the popular SVM (Vapnik, 1998). A finer analysis indicates that the second error term of the total risk's upper bound is increasing when the Vapnik-Chervonenkis (VC) dimension $\hbar$ is also increasing. This latter is directly linked

with the particular choice of the kernel, $\hbar = 2K+1$ for linear kernel and $\hbar = \infty$ for RBF kernel. In other words, one may prefer a simple linear separator especially when the input feature dimension K is high. It will generally perform close to those obtained with a non-linear specialized kernel but with faster training and prediction.

SVM is trained classically with the Sequential Minimal Optimization (SMO) algorithm (Schölkopf et al., 2001) with a complexity $O(KP^2)$ where P is the number of training examples (in the worst case, when $P_{sv} = P$, P_{sv} is the number of support vectors). This quadratic dependency from P can be reduced if we particularize kernels to linear ones and by introducing an extra tolerance term ε in the minimization problem. This leads to large-scale linear SVM solvers based on efficient Newton optimizers (Hsieh et al., 2008) or stochastic gradient descent (on the primal form, see (Shalev-Shwartz et al., 2007)). Complexity of such a large-scale solver is reduced to $O(KP)$.

Let us define the set of sparse code descriptors (after mean- or max-pooling) $Z \triangleq \{z_1, \ldots, z_P\}$ and their corresponding labels $y \triangleq \{y_1, \ldots, y_P\}$ where $z_i \in \mathcal{Z} \subseteq \mathbb{R}^K$ and $y_i \in \{-1, 1\}$. Thus, each phoneme recognition is considered as a class/non-class problem with a one-against-all approach. P corresponds to the number of phoneme realizations, which is inferior to the total number N of MFCC parameter vectors and equals the number of descriptors after pooling method.

The linear SVM problem consists of finding the hyperplane parameter $\widehat{w}$ minimizing the sum of a ℓ_2 loss function and a ℓ_2 regularization term such that:

$$\widehat{w}^T = \arg\min_{w} \left\{ \frac{1}{2} w^T w + C \sum_{i=1}^{M} \max\left(1 - y_i w^T z_i, 0\right)^2 \right\} \quad (9)$$

To solve this problem, we use a modified version of the liblinear 1.5 package (Fan et al., 2008) accepting dense input vectors [1]. Any new input vector y will be classified as $\widehat{z} = \text{sign}(f(z;\widehat{w})) = \text{sign}(\widehat{w}^T z) \in \{-1, 1\}$ by a simple scalar product in $O(K)$.

6.1 Fast Non-Linear Method for Approximated Additive Homogeneous Kernels via Explicit Feature Maps

For linear SVM, evaluation is performed with a simple scalar product $f_l(z;w) = w^T z$ taking $O(K)$ whereas for non-linear kernel the expansion becomes $f_{nl}(z;\beta) = \sum_{i=1}^{P_{sv}} \beta_i K(z, z_i)$ taking approximately $O(KP_{sv})$. Both for training and predicting, SVM with non-linear kernels is at least P_{sv} slower.

First introduced by (Maji et al., 2009) for the intersection histogram (IH) kernel and extended to any homogeneous additive kernels (IH, χ^2, Shannon-Jensen) in (Vedaldi and Zisserman, 2011), an explicit closed form of the feature maps associated to these kernels, denoted $\Psi(z)$, permits the approximation $f_{nl}(z) \approx w'^T \Psi(z) = w'^T z'$. Moreover the $\Psi(z)$ approximation is independent of the training data. Now w' and $z' \in \mathbb{R}^{(2v+1)K}$ where v is the approximation order (typically $v = \{1, 2\}$) and $f_{nl}(z)$ runs in $O((2v+1)K)$.

Assuming z_j is the j^{th} components of z, $j = 1, \ldots, K$ then for any homogeneous kernel, $\Psi(z_j)$ is approximated as follows:

$$\widehat{\Psi}_i(z_j) = \begin{cases} \sqrt{\widehat{\kappa}_0} & i = 0, \\ \sqrt{2 z_j \widehat{\kappa}_{\frac{i+1}{2}}} \cos\left(\frac{i+1}{2} L \log z_j\right) & i > 0 \text{ odd}, \\ \sqrt{2 z_j \widehat{\kappa}_{\frac{i}{2}}} \sin\left(\frac{i}{2} L \log z_j\right) & i > 0 \text{ even}, \end{cases} \quad (10)$$

$\widehat{\kappa}_i$ is the i^{th} value of the spectrum $\kappa(\omega)$ sampled with a sampling frequency equal to L. Closed forms of $\kappa(\omega)$ depend on the chosen kernel type and readers can retrieve details in (Vedaldi and Zisserman, 2011). For an efficient implementation, Vedaldi also proposes to pre-compute values of $\widehat{\Psi}_i(z_j)$ for wide dynamic of z_j and store them in a table. With this approach, retrieving $\widehat{\Psi}_i(z_j)$ with this given table runs in $O(2v+1)$.

Since it takes an extra $O((2v+1)K)$ to compute $\Psi(z)$ from z, total run time for using an additive homogeneous kernel is $O(2(2v+1)K)$ ($\Psi(z)$+$f_{nl}(z)$ computation) and even more important is independent of P_{sv} and consequently of the training size P. The acceleration is proportional to $\frac{P_{sv}}{2(2v+1)}$ for the non-linear case.

To make experimentations, we used a modification of the Scenes/Objects Classification Matlab toolbox framework which implements all the above mentioned processing (Paris, 2011).

7 CORPUS DESCRIPTION

To develop our method, we used about 2 hours of French radio broadcast news. This corpus was extracted from a larger broadcast news corpus provided by the 2006 ESTER French evaluation campaign (Gravier et al., 2004; Galliano et al., 2006; Razik et al., 2011). This corpus contains only broadband speech (no narrow band, no music segments) and sentence level transcriptions are provided. However, some sentences may have background noise or

[1] Available at `http://www.cs.berkeley.edu/?smaji/projects/digits/`

music. There are 72 speakers in the 2 hours, 28 females and 44 males. This corpus is more difficult and more realistic (noise, number of classes, etc.) than the usual TIMIT data set.

In our study, we work at phoneme level, so we used a forced alignment process of the data by external acoustical models and lexicon of a large vocabulary speech recognition systems (Illina et al., 2004). We used a decomposition on $V = 41$ french phonemes including silence and short pause. Figure 1 shows the histogram of the phoneme distribution within the 2 hours of our corpus.

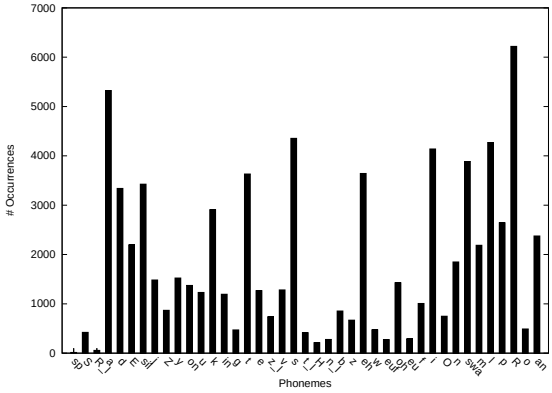

Figure 1: Phoneme distribution in the corpus.

As with this corpus some phonemes are underrepresented, we will use a cross validation technique in 10 subsets to assess the performance of both developed systems instead of fixed training, development and test corpora. One of the ten subset was used as a development corpus to tune the hyperparameter of the SVM.

The subsets were not defined or clustered according to the speakers thus forthcoming results are speaker independent.

According to the corpora size, the different results are given in recognition rate and standard deviation (vertical bars in figures), and to a significance confidence of around 0.2% at 0.95 level of significance.

8 RESULTS

As the corpus is labeled in 41 phoneme classes, we learn 41 GMM models (one model for each phoneme) with various number of components. We use the HTK toolkit (Young et al., 1995) for extracting both the MFCC coefficients and training the GMM models[2].

[2]However we do not use the Hidden Markov Model (HMM) capability of HTK and trained our GMM by a one state HMM.

We assess a number of Gaussian mixtures varying between 1 to 256. Figure 2 shows that the GMM system obtained a best performance of about 61% of accuracy for 100 mixtures on our corpus. This performance is of good level knowing that our corpus can be considered difficult (only 1h of training data, broadcast radio condition).

8.1 Whitening The Data

Additionally to the comparison between GMM and SC/SVM model, we assess the use of a whitening preprocessing of the MFCC data as explained in section 2.1.

Figure 2 shows the accuracy rate of the GMM phoneme recognition system according to the number of mixtures for both raw and whiten MFCC data. As for the case of image processing (Coates et al., 2011), whitening the data before computing both GMM or Sparse Dictionary improves the results by around 1% in absolute.

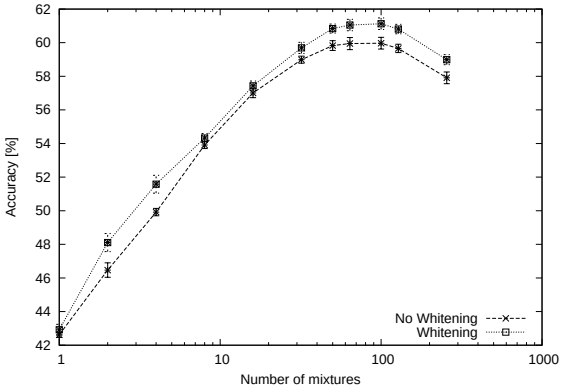

Figure 2: Effect of whitening with GMM according to the number of mixtures (log scale).

8.2 Sparse Coding/SVM

In the SC/SVM based system, we assess the role of several parameters on the accuracy as the choice of the pooling method, the use of linear or non-linear kernel (intersection kernel) for the SVM and of course the size of the dictionary.

Concerning the recognition system process, the SVM hyper-parameter is tuned globally on all phoneme classes and not specifically to each class. Moreover, the descriptor vectors (sparse codes) are normalized according to the kernel type. In the linear case, the vectors should be ℓ_2 normalized, and in the

non-linear case, the vectors should be ℓ_1 normalized (Vedaldi and Zisserman, 2011).

In our experiment, the system obtains the same accuracy performance whatever which pooling method is used.

As we mentioned, the use of an approximated non-linear kernel SVM improves the accuracy of our system compared to the linear case. Figure 3 shows the accuracy of both linear and intersection kernels. We can note that the approximated non-linear system performs better than the linear one. Even if their accuracy are close, the difference between both is significant and the training time is just slightly increased by $2\nu + 1$ where $\nu = 1$ in our study.

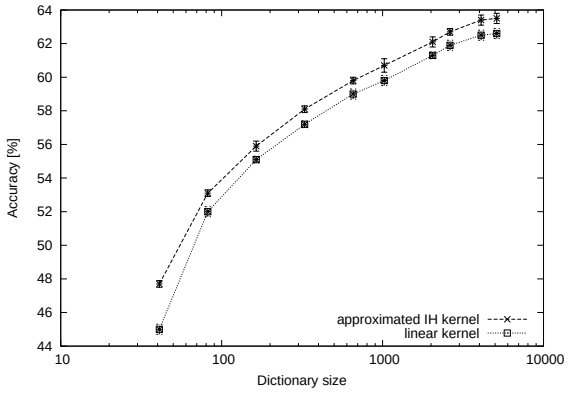

Figure 3: Accuracy with linear and approximated IH kernel ($\nu = 1$) in log scale.

8.3 GMM vs. Sparse Coding

We compare both GMM and Sparse Coding based systems on the corpus. In order to compare their results, we consider that a GMM with M mixtures is similar to a codebook dictionary with $K = 41 \times M$ vectors (number of classes by number of mixtures).

Figure 4 shows the accuracy of both system according to this comparison scale. The SC/SVM based system outperforms the GMM based system by around 2.5% in absolute (63.5% for the Sparse Coding/SVM and 61.1% for the GMM) at their best.

9 CONCLUSIONS AND PERSPECTIVES

In this paper we showed that the sparse coding approach outperforms significantly the classic GMM. For any number of Gaussian components, the system

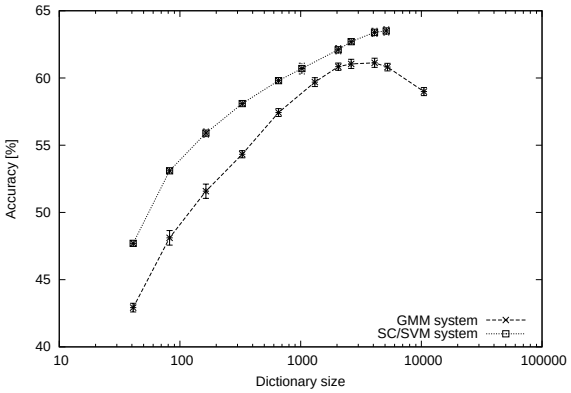

Figure 4: Accuracy of SC/SVM vs. GMM (log scale).

using the equivalent number of sparse codes outperforms the GMM. An advantage of the sparse coding method is that it needs less training samples to learn a representative dictionary and the representation is more compact. The GMM method is more sensitive to the total amount of available data and to the initialization step. Moreover, in this paper we achieved results close to a specialized non-linear kernel thanks to the additive homogeneous kernel approximation, still linear in computation time.

Several directions could be explored to further improve the performance of the sparse approach, including Laplacian constraints in the sparse codes construction to obtain a more reliable codebook versus a slight change in the data (Wang et al., 2010), using a hierarchical construction of the codebook (Yu et al., 2011), or providing a feature selection method based on MKL (Multiple Kernel Learning) (see FGM – Feature Generating Machine algorithm (Tan et al., 2010)). Although we used an unsupervised method for building the codebook, it is also possible to train simultaneously the dictionary and the classifier in a supervised way (Mairal et al., 2008).

Furthermore, the results obtained in this study are only focused on the phone stage of speech recognition. We should evaluate the impact of the improvement at this level on the final word recognition rate of a complete ASR system. However, the ASR system can still be based on an HMM classifier with the observation matrix directly build with the probability outputs of the SVM. It has the advantage of being a method both generative and discriminative. Finally, as it has been introduced more recently, it is possible to use a structural SVM learning discriminately and simultaneously the temporal structure and the sparse codes in linear time (Joachims et al., 2009).

REFERENCES

Arous, N. and Ellouze, N. (2003). Cooperative supervised and unsupervised learning algorithm for phoneme recognition in continuous speech and speaker-independent context. *Neurocomputing*, 51:225–235.

Coates, A., Lee, H., and Ng, A. Y. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Artificial Intelligence and Statistics (AISTATS)*, page 9.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. ASSP*, 28:357–366.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J., Mostefa, D., and Choukri, K. (2006). Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *LREC*, pages 315–320.

Gravier, G., Bonastre, J., Galliano, S., and Geoffrois, E. (2004). The ester evaluation compaign of rich transcription of french broadcast news. In *LREC*.

Hammer, B., Strickert, M., and Villmann, T. (2004). Relevance lvq versus svm. In *Artificial Intelligence and Softcomputing, springer lecture notes in artificial intelligence*, volume 3070, pages 592–597. Springer.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., and Keerthi, S. S. (2008). A dual coordinate descent method for large-scale linear svm.

Huang, X., Acero, A., and Hon, H. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Netw.*, 13:411–430.

Illina, I., Fohr, D., Mella, O., and Cerisara, C. (2004). The automatic news transcription system : Ants, some real time experiments. In *ICSLP*, pages 377–380.

Joachims, T., Finley, T., and Yu, C.-N. (2009). Cutting-plane training of structural svms. *Machine learning*, 77(1):27–59.

Lin, C.-J., Weng, R. C., and Keerthi, S. S. (2008). Trust region newton method for logistic regression. *J. Mach. Learn. Res.*, 9.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA. ACM.

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

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Supervised dictionary learning. *Advances Neural Information Processing Systems*, pages 1033–1040.

Maji, S., Berg, A. C., and Malik, J. (2009). Classification using intersection kernel support vector machines is efficient. In *CVPR*.

Paris, S. (2011). Scenes/objects classification toolbox. `http://www.mathworks.com/matlabcentral/fileexchange/29800-scenesobjects%-classification-toolbox`.

Rabiner, L. and Juang, B. (1993). *Fundamentals of Speech Recognition*. Prentice Hall PTR.

Ranzato, M., Krizhevsky, A., and Hinton, G. (2010). Factored 3-way restricted boltzmann machines for modeling natural images. In *International Conference on Artificial Intelligence and Statistics AISTATS*.

Razik, J., Mella, O., Fohr, D., and Haton, J.-P. (2011). Frame-synchronous and local confidence measures for automatic speech recognition. *IJPRAI*, 25(2):157–182.

Rudin, C., Schapire, R. E., and Daubechies, I. (2007). Analysis of boosting algorithms using the smooth margin function. *The Annals of Statistics*, 35(6):2723–2768.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13:1443–1471.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2007). Pegasos: Primal estimated sub-gradient solver for svm.

Sivaram, G., Nemala, S., M. Elhilali, T. T., and Hermansky, H. (2010). Sparse coding for speech recognition. In *ICASSP*, pages 4346–4349.

Smit, W. J. and Barnard, E. (2009). Continuous speech recognition with sparse coding. *Computer Speech and Language*, 23:200–219.

Tan, M., Wang, L., and Tsang, I. W. (2010). Learning sparse svm for feature selection on very high dimensional datasets. In *ICML*, page 8.

Tomi Kinnunen, H. L. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52:12–40.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Intersciences.

Vedaldi, A. and Zisserman, A. (2011). Efficient additive kernels via explicit feature maps. *IEEE PAMI*.

Wang, J., Yang, J., Kai Yu, F. L., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. *CVPR'10*.

Yang, J., Yu, K., Gong, Y., and Huang, T. S. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1995). *The HTK Book*. Entropic Ltd., Cambridge, England.

Yu, K., Lin, Y., and Lafferty, J. (2011). Learning image representations from the pixel level via hierarchical sparse coding. In *CVPR*, pages 1713–1720.