TOWARD A PRACTICAL IMPLEMENTATION OF EXEMPLAR-BASED NOISE ROBUST ASR

Jort F. Gemmeke[1], *Antti Hurmalainen*[2], *Tuomas Virtanen*[2], *Yang Sun*[1]

[1]Department of Linguistics, Radboud University, Nijmegen, The Netherlands.
[2]Department of Signal Processing, Tampere University of Technology, Finland.
jgemmeke@amadana.nl antti.hurmalainen@tut.fi tuomas.virtanen@tut.fi y.sun@let.ru.nl

ABSTRACT

In previous work it was shown that, at least in principle, an exemplar-based approach to noise robust ASR is possible. The method, sparse representation based classification (SC), works by modelling noisy speech as a sparse linear combination of speech and noise exemplars. After recovering the sparsest possible linear combination of labelled exemplars, noise robust posterior likelihoods are estimated by using the weights of the exemplars as evidence of the state labels underlying exemplars. Although promising recognition accuracies at low SNRs were obtained, the method was impractical due to its slow execution speed. Moreover, the performance was not as good on noisy speech corrupted by noise types not represented by the noise exemplars. The importance of sparsity was poorly understood, and the influence of the size of the exemplar-dictionary was unclear. In this paper we investigate all these issues, and we show for example that speedups of a factor 28 can be obtained by using modern GPUs, bringing its execution speed within range to practical applications.

1. INTRODUCTION

For the last 30 years Automatic Speech Recognition (ASR) has been completely dominated by the use of Hidden Markov Models (HMMs). HMM-based ASR performance, however, degrades substantially when speech is corrupted by background noise not seen during training. Additionally, it has become clear that not all speech phenomena can be covered in the form of HMMs. There is a general agreement in the speech community about the need for novel approaches for handling phenomena that HMMs do not account for (cf. [1] and the references therein).

In [2] a new approach to noise robust speech recognition was introduced. The approach, dubbed *sparse classification* (SC), is an exemplar-based approach based on the idea that speech signals can be represented as a sparse, non-negative linear combination of a small set of suitably selected exemplars. First the linear combination is recovered by finding the smallest number of labelled exemplars in a very large collection of exemplars (a *dictionary*) that *jointly* approximates the observed speech signal. After obtaining the sparse representation, the weights of the linear combination of exemplars are used together with their associated (HMM-state) labels to provide state likelihoods, after which recognition is done using Viterbi decoding. It was proposed to make SC noise robust by modelling noisy speech as a linear combination of speech and noise exemplars. It was shown that promising accuracies at low SNRs can be reached, especially for matched noise types. In [3] refinements to SC were proposed that improved performance at high SNRs.

In this paper, we address important issues that stand in the way of a practical implementation of SC. The first issue is computational complexity: Due to its exemplar-based nature and the need to find a sparse linear combination, SC is too slow for practical applications. The method, however, lends itself well to parallelisation due to its reliance on linear algebra operations on large matrices. It has been shown that modern hardware, such as Graphics Processing Units (GPUs) can dramatically accelerate linear algebra calculations [4]. One drawback however, is that most of these speedups can only be obtained when using single precision floating point variables, rather than double precision variables. In this work, we investigate if there is an impact of using single precision floating point variables, and what kind of speedups can be obtained when using GPUs to accelerate SC.

The second issue is the sensitivity to the choice of noise dictionary. Previous research has shown that while SC achieves impressive noise robustness when the noise types of the available noise exemplars correspond to the noise types encountered in the noisy speech, the method does not generalise as well to unseen noise types. In [5], it was proposed to use artificial noise exemplars to cover such unseen noises, and it was shown an artificial noise dictionary achieves better noise robustness on unseen noise types. In this paper, we explore to what extend the good performance on both seen and unseen noise types can be retained when the the two noise exemplar types are combined in one system.

Third, the SC method relies on finding a *sparse* linear combination of exemplars. While experiments revealed that the use of a sparsity-inducing penalty in the cost function that is minimised to obtain a sparse representation is beneficial, the relation between sparsity and recognition accuracy remained unclear. In this work, we investigate the influence of sparsity in more detail. Finally, there is the issue of choosing the size of the speech exemplar dictionary. In previous works [2, 3] only a single dictionary size was used, although it was stated that larger dictionary sizes could improve recognition accuracy. In this work, we explore to what extend the recognition accuracy is dependent on the dictionary size, both in clean conditions and in noisy conditions.

2. SPARSE CLASSIFICATION

2.1 Sparse representation of noisy speech

In ASR, speech signals are represented by their spectro-temporal distribution of acoustic energy, a *spectrogram*. The magnitude spectrogram describing a clean speech segment $\mathbf{S}$ is a $B \times T$ dimensional matrix (with B frequency bands and T time frames). To simplify the notation, the columns of this matrix are stacked into a single vector $\mathbf{s}$ of length $D = B \cdot T$.

We assume that an observed speech segment can be expressed as a sparse, linear, non-negative combination of clean speech exemplars $\mathbf{a}^{\mathrm{s}}_j$, with $j = 1, \ldots, J$ denoting the exemplar

index. We model noise spectrograms as a linear combination of noise exemplars $\mathbf{a}^{\mathrm{n}}_k$, with $k = 1,\ldots,K$ being the noise exemplar index. This leads to representing noisy speech $\mathbf{y}$ as a linear combination of both speech and noise exemplars:

$$\mathbf{y} \approx \mathbf{s} + \mathbf{n} \tag{1}$$

$$\approx \sum_{j=1}^{J} x^s_j \mathbf{a}^{\mathrm{s}}_j + \sum_{k=1}^{K} x^{\mathrm{n}}_k \mathbf{a}^{\mathrm{n}}_k \tag{2}$$

$$= [\mathbf{A}^{\mathrm{s}} \mathbf{A}^{\mathrm{n}}] \begin{bmatrix} \mathbf{x}^{\mathrm{s}} \\ \mathbf{x}^{\mathrm{n}} \end{bmatrix} \tag{3}$$

$$= \mathbf{A}\mathbf{x} \quad \text{s.t.} \quad \mathbf{x}^{\mathrm{s}}, \mathbf{x}^{\mathrm{n}}, \mathbf{x} \geq 0 \tag{4}$$

with $\mathbf{x}^{\mathrm{s}}$ and $\mathbf{x}^{\mathrm{n}}$ sparse representations of the underlying speech and noise, respectively. In order to obtain $\mathbf{x}$, we minimise the cost function:

$$d(\mathbf{y}, \mathbf{A}\mathbf{x}) + ||\lambda \ .* \mathbf{x}||_1 \quad \text{s.t.,} \quad \mathbf{x} \geq 0 \tag{5}$$

with as distance function d the generalised Kullback-Leibler (KL) divergence and with the second term a sparsity inducing L-1 norm of the activation vector weighted by element-wise multiplication (operator $.*$) with vector $\lambda = [\lambda_1 \ \lambda_2 \ldots \lambda_L]$. The cost function (5) is minimised using a multiplicative updates routine as in [2].

2.2 Noise robust likelihoods

In order to decode utterances of arbitrary lengths, we adopt a sliding time window approach as in [2]. In this approach, we represent a noisy utterance as W fixed-size, overlapping speech segments. For each segment, we calculate a sparse representation as described above. With each frame within each exemplar in the speech dictionary labelled using HMM-state labels obtained from a conventional MFCC-based decoder, the weights of the exemplars in the sparse representation are directly used to calculate posterior state likelihoods for that segment using the method described in [3]. We obtain a likelihood matrix for the entire utterance by averaging the likelihoods of the frames of all the segments that overlap, and decode the speech utterance by using a Viterbi search for the state sequences which maximise likelihood.

3. EXPERIMENTAL SETUP

For our recognition experiments we used material from test sets 'A' and 'B' of the AURORA-2 corpus. Test set A comprises 1 clean and 24 noisy subsets, containing four noise types (subway, car, babble, exhibition hall). For testing we used these same random, representative subset of 10% of the utterances (i.e. 400 utterances per SNR level) used in [2]. In the results, only three SNR values, 5 and -5 dB as well as clean speech, are shown to improve the clarity of the results. Test set B contains four different noise types (restaurant, street, airport, train station). In the experiments, the results of the four noise types are averaged. Each subset contains 1001 utterances with one to seven digits '0-9' or 'oh'. Acoustic feature vectors consisted of mel frequency magnitude spectrograms, spanning $T = 23$ bands with a frame length of 25 ms and a frame shift of 10 ms.

The speech and noise dictionaries were created in a two-step procedure which is repeated for each exemplar size $T \in \{20, 30\}$. First, from each noisy utterance in the multi-condition AURORA-2 training set two segments were randomly selected. The segments were allowed to overlap and no effort was made to exclude silence frames from the exemplars. For these segments the underlying clean speech and noise originally used for creating the noisy speech were extracted from their respective spectrograms and added to the speech and noise dictionaries. This resulted in initial speech and noise dictionaries consisting of 16880 exemplars. Unless stated otherwise, the speech and noise dictionaries consisted of 4000 exemplars randomly chosen from the initial speech and noise dictionaries. The multi-condition training set of AURORA-2 contains the same noise types as test set A.

Digits were described by 16 states with an additional 3-state silence word, resulting in a 179 dimensional state-space. The speech decoding system was implemented in MATLAB; we refer the reader to [3] for details. The machine used for the experiments was a Core 2 Duo E6750 2.4 GHz, with 4 GB of RAM. It was equipped with a 1GB GTX460 GPU and operated under Windows 7, 64-bit with Matlab2010b.

4. EXPERIMENTS AND RESULTS

4.1 GPU-based acceleration

Modern multi-core hardware, such as GPUs, can dramatically improve the speed at which linear algebra calculations are carried out (e.g. [4] and the references therein). Up until quite recently, however, writing programs which utilise the GPU was a difficult and time-consuming task. Now, a number of software packages exist for high-level languages such as MATLAB which enable the programmer to quickly develop programs which can be accelerated by the GPU. In this paper we use one of these packages, the freeware MATLAB toolbox GPUmat [6]. This toolbox allows GPU-based calculation simply by first initialising the variables used in a calculation for use on the GPU, after which calculation of the involved variables is automatically carried out on the GPU without further modification of the MATLAB code.

The GPU based acceleration is most effective when using single precision variables. In this first experiment, we investigate whether the use of single precision floating point variables has a significant impact on the accuracies obtained with SC. We test this by replacing the multiplicative update routine which solves (5) by a function which first converts the dictionary and speech segments to single precision for use on the GPU. It then finds the sparse representation using the GPU and finally converts the resulting sparse representation vector $\mathbf{x}$ back to double precision for the calculation of noise robust likelihoods on the CPU.

In Table 1 we show the results using test set A. From the results we can observe that there is no significant (assuming a binomial random variable and a 95 % confidence interval) impact on recognition accuracy from using single precision GPU-based calculations. To characterise the computational effort needed, we did timing experiments using the utterance 'MIP_68385A', taken from test set A, subway noise type, SNR -5 dB, which has a length of 182 frames (1.82 seconds of speech). The time spent on transferring data from and to the GPU (a costly operation) is included in the timings. As the running time of the minimisation of cost function (5) is data-independent, averaging over repeated runs on the same file is representative for the complete dataset.

It can be inferred from the results in Table 2 that using the GPU results in a speedup of a factor 24 and 28 for $T = 20$ and $T = 30$, respectively. The fact that larger exemplar sizes (containing slightly less segments which need to be processed but more variables per segment) achieve a larger speedup, is mostly due to the fact multiplications of large

Table 1: Comparison of word recognition accuracy between GPU and CPU. The results pertain test set A.

SNR [dB]		clean	5	-5
T=20	CPU	96.8	88.7	51.9
	GPU	96.8	88.7	51.9
T=30	CPU	94.6	89.9	**57.3**
	GPU	94.6	89.9	57.2

Table 2: Average running times of a single utterance for different exemplar sizes. The 'solver' column depicts the time spent on minimising update rule (5).

		CPU		GPU	
	W	solver [s]	total [s]	solver [s]	total [s]
T=20	163	65.2	65.4	2.6	2.7
T=30	153	92.1	92.2	3.3	3.4

matrices can be parallelised better and thus larger speedups on a GPU. Additionally, there is the relative cost of transferring data to and from the GPU. It is therefore expected that even larger speedups can be obtained by processing several utterances at once if the utterances are short, but also by doing (part of) the likelihood calculation on the GPU. Especially considering that the employed GPU can be considered a 'mainstream' model, the obtained speedups show real-time processing using SC-based methods is now well within reach. We will use the GPU-based solver in the remainder of the experiments.

4.2 Combination of noise dictionaries

In [5], it was proposed to use artificial noise exemplars rather than 'real' noise exemplars extracted from a noise database in order to provide robustness against unseen noise types. In that work, a very simple form of artificial exemplars was proposed: the use of $B = 23$ exemplars that each have one non-zero, constant frequency band. It was shown that using these already improves the accuracy on unseen noise types. At the same time, the artificial exemplars are not as noise robust as real exemplars if the noise types match.

In realistic applications, it can be assumed that there is a rough idea about the noise types that can be expected in the observed noisy speech, while at the same time the speech recogniser should be as robust as possible against unseen noises. In this work, we mimic this scenario and explore whether the two forms of noise dictionaries - real and artificial - can be effectively combined.

In Table 3, the results are shown for the use of only real, only artificial and both types ("combined") of noise exemplars. Test set A contains the same noise types as those in the real noise exemplar dictionary (matched test set) while test set B contains unseen noise types (mismatched test set). As was observed in [5], the use of the artificial noise exemplars leads to a lower performance on the matched test set when using $T = 30$ but better on the mismatched test set. When using $T = 20$, the use of the artificial noise dictionary degrades the results in all noisy conditions.

When combining the two noise dictionaries (keeping all other parameter settings unchanged), the results on the mismatched test set improve for both the real and the artificial noise dictionaries, also for $T = 20$. On the matched test set, the use of the combined system still results in a small performance degradation, although far less than when using artificial noise exemplars.

Table 3: Comparison of noise exemplar types. Recognition accuracies for two exemplar sizes, T=20 and T=30 are shown, both in test set A and test set B. The rows labelled 'real' and 'artificial' refer to the two types of noise exemplars, with the 'combined' system employing both.

		test set A			test set B	
SNR [dB]		clean	5	-5	5	-5
T=20	real	96.8	88.7	51.9	82.4	33.8
	artificial	96.8	76.8	26.7	76.6	30.5
	combined	96.9	87.0	47.2	89.3	53.2
T=30	real	94.6	89.9	57.2	84.8	37.8
	artificial	95.3	84.9	39.2	86.5	45.3
	combined	94.6	89.3	53.2	88.5	49.3

The reason for the improvement of the combined system over the purely artificial exemplars on the mismatched dataset can be found in the way silence states are treated in the system. As explained in [3], the silence states do not get activated during silence because an absence of speech energy is modelled as all-zeros sparse representation. As a work-around, silence states are activated based on the balance between noise and speech exemplar activations. The hyper-parameters governing this process are guided by an SNR estimator, which is also based on the balance between noise and speech exemplar activations. Preliminary investigations seem to indicate that it is this SNR estimation that is no longer adequately tuned when using an artificial noise dictionaries. With the addition of the real exemplars, the SNR estimation works properly again, leading to improvements over the individual systems.

From these results we can conclude that exemplar noise dictionaries can be combined to achieve robustness against both known and unseen noise types, although the way silence states are treated should be modified to become less dependent on the choice of noise dictionary. One possible way to do this is by using an separate SNR estimator that works directly on the noisy speech.

4.3 The importance of sparsity

As the name implies, Sparse Classification (SC) relies on finding a *sparse* linear combination of exemplars. In [2] it was stated that sparsity for speech was important because it avoids over-fitting of the representation $\mathbf{x}$ and forces the exemplars that are selected to be closer to the underlying, lower-dimensional, manifolds on which the various digit or state classes are located.

In practice, the sparsity of the combination of exemplars is regulated through the use of a sparsity inducing norm governed by the λ variable. The cost function (5) allows for the specification of a different sparsity-inducing weight for any part of the dictionary, but in [2] only the sparsity of the speech exemplars was enforced and the weights λ for the noise part of the dictionary were set to zero.

In order to investigate the influence of the sparsity weight, and the effect of having a non-zero sparsity weight for noise exemplars, we investigated recognition accuracy as a function of sparsity using a small subset of randomly selected SNR 5 dB utterances of the multi-condition train set. In Figure 1, we show the recognition accuracy as a function of a single sparsity weight for the speech part (denoted λ_s) and the noise part (denoted λ_n) of the dictionary, for $T = 20$.

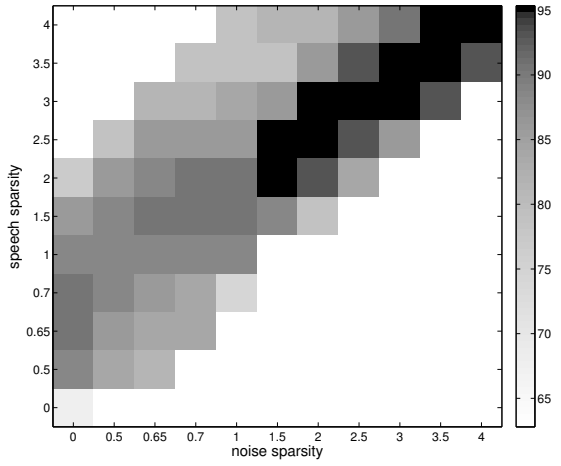

Figure 1: Word recognition accuracy on SNR 5 dB utterances of the multi-condition training set for various values of the sparsity weight for the speech part and the noise part of the dictionary. Areas that are white have not been evaluated.

It can be observed that with $\lambda_n = 0$, the optimal value for λ_s is 0.65, which is indeed the value reported in [2] and used elsewhere in this paper. When considering non-zero values for λ_n, however, it can be seen there is an strong correlation between the optimal speech and noise sparsity. As a rule of thumb, it seems that the optimal values are just off-diagonal, with the speech sparsity slightly higher than the noise sparsity. It can also be observed that the range of recognition accuracies is quite large, from 63% at when no sparsity is enforced ($\lambda_y = 0, \lambda_n = 0$), to 95%, for ($\lambda_y = 2, \lambda_n = 2$).

To see how well tuned sparsity values found in this manner will generalise to the test set, we do a new recognition experiment using $\lambda_y = 2, \lambda_n = 1.5$. The results in Table 4 show that the recognition accuracies are improved in SNR 5 dB, which is the SNR on which was tuned, but also on clean speech. In the SNR -5 dB condition, however, we observe a decrease in accuracy. It is worth noting that also on the unseen noise types of test set B, the performance of the tuned system improves at SNR 5 dB.

Further investigation of the influence of the sparsity at various SNR's (not shown) revealed that the range of sparsity values with which the highest accuracies are obtained (black areas in Figure 1) get smaller at lower SNRs, while the absolute values of the sparsity weights increase. In other words, in the presence of strong background noise it becomes beneficial to require sparser solutions, which makes the separation of speech and noise easier, but those same values can hurt performance at high SNRs.

From this we can conclude that enforcing sparsity does indeed improve the performance of SC, but also that the choice of sparsity weight depends on the type of exemplars (speech or noise) dictionary and the SNR. As no (non-Bayesian) method of enforcing sparsity for cost functions of the form (5) is known that does not require explicit or implicit tuning of a sparsity-governing parameter, means we should find smarter ways to govern the sparsity weight λ. It should be possible to train a set of SNR-dependent parameters in combination with a SNR estimator, similar to the method now used for balancing silence states in SC (cf. [3]).

Table 4: Comparison of word recognition accuracy as a function of the optimisation of sparsity weights.

SNR [dB]		test set A clean	test set A 5	test set A -5	test set B 5	test set B -5
T=20	Baseline	96.8	88.7	**51.9**	82.4	**33.8**
	Tuned	**97.4**	**90.3**	41.6	**87.5**	33.6
T=30	Baseline	94.6	89.9	**57.2**	84.8	**37.8**
	Tuned	**96.8**	**91.2**	53.1	**87.8**	34.2

4.4 Influence of dictionary size

Databases employed in ASR can contain tens of millions frames of speech data. Especially for exemplar-based methods, this poses a problem as ideally all the resulting millions of exemplars should be considered during recognition. To make recognition computationally feasible, in the experiments with SC on AURORA-2 a speech dictionary of only 4000 speech exemplars is created by random sampling. This procedure was outlined in Section 3.

The use of random sampling gives to rise to the question how sensitive the performance of SC is as a function of dictionary size. In [7], the influence of dictionary size on recognition accuracy was first investigated. It was shown that larger dictionary sizes resulted in higher recognition accuracies, although the returns were diminishing. That work however, employed a predecessor of SC: log-spectral features and a euclidean distance criterion were used, and since no noise dictionary was used, only results on clean speech were reported.

In this section, we explore the influence of dictionary size on clean speech as well as matched and mismatched noisy speech. Starting from the initial speech dictionary of 16880 exemplars (reduced to 16000 for convenience) described in Section 3, we create several smaller sized dictionaries: $J \in 250, 500, 1000, 2000, 4000, 8000$. The procedure for creating these is simply that each dictionary size is randomly split into two dictionaries half the size. Since the choice of random subset might influence the results, we average the results over four different, non-overlapping subsets for dictionary sizes $J < 4000$. A separate experiment (not shown) showed that for $J \geq 4000$, performance did not differ more than 0.1 percent (absolute) when taking a different subset of the initial dictionary. Throughout the exemplars, the same noise dictionary of size $K = 4000$ was used.

The results on $T = 30$ in Table 5 reveal that the clean speech performance rises from 94.6% to 95.9% when increasing the number of speech exemplars from 4000 to 16000. In general, the performance both for clean speech and for noisy speech seems to increase as the dictionary size increases, although on SNR -5 dB the accuracy no longer increases if $K > 4000$. It is worth noting that when using very small dictionary sizes, the performance on matched noisy speech is already impressive with 40.6% at SNR -5 dB.

The lack of improvement (and even degradation) on matched noisy speech observed for $K > 4000$, is surprising however. Investigation of this phenomenon revealed that this is due to the way (5) is solved. The cost function (5) is minimised by a multiplicative updates routine as described in [2]. So far, that multiplicative updates algorithm operates with a fixed number of iterations, $F = 200$. The number of iterations that was used, however, was set by a non-exhaustive investigation of the convergence of the cost function that is minimised.

Table 5: Word recognition accuracy as a function of the number of exemplars in the speech part of the dictionary. In these experiments, $T = 30$ was used. Reported accuracies are averages over multiple choices

	test set A			test set B	
SNR [dB]	clean	5	-5	5	-5
$J = 250$	79.6	71.3	40.6	67.9	29.0
$J = 500$	86.9	78.7	47.2	75.0	32.3
$J = 1000$	90.7	83.8	52.9	79.2	34.4
$J = 2000$	92.7	86.9	55.2	82.3	36.1
$J = 4000$	94.6	89.9	57.2	84.8	37.8
$J = 8000$	95.4	89.4	56.2	84.5	36.1
$J = 16000$	95.9	90.1	55.5	84.2	38.1

Table 6: Word recognition accuracy as a function of the number of iterations. The results pertain test set A.

SNR [dB]		clean	5	-5
T=20	$F = 200$	96.8	88.7	51.9
	$F = 250$	96.8	88.9	54.2
	$F = 300$	96.8	**89.0**	**54.3**
T=30	$F = 200$	94.6	89.9	57.2
	$F = 250$	**94.8**	90.4	60.3
	$F = 300$	94.8	**90.6**	**62.4**

In Table 6, we show recognition accuracy as a function of the number of iterations and exemplar size. The results (using $K = 4000$) show that for $T = 20$, higher iteration counts have no impact on the clean speech recognition accuracy, but have a significant effect at lower SNRs: the accuracy increases from 51.9 to 54.3% accuracy at SNR -5. For $T = 30$, higher iteration counts also have a positive, although not significant effect on clean speech accuracy, and the accuracy increases from 57.2 to 62.4% accuracy at SNR -5. Results on test set B (not shown) reveal that the performance on the mismatched noises in test set B also increases significantly, although less than on test set A. Iteration counts higher than $F = 300$ did not lead to further improvements.

A repeat of the dictionary-size experiment with $F = 300$ (not shown) showed that the performance on SNR -5 dB matched noisy speech was constant at $K > 4000$, while on the mismatched test set and at all other SNRs the performance kept increasing with dictionary size. From these experiments we can conclude that the SC performance reported thus far was suboptimal, and that in general, the bigger the speech dictionary, the better the performance.

At the same time, the computational complexity increases linearly with dictionary size, so for a practical application of SC a more principled way of constructing the speech dictionary would be preferable. Unfortunately, our preliminary experiments employing clustering techniques did not result in dictionaries that work better than dictionaries randomly extracted from the speech database. Perhaps a more promising way would be to find a way to use hierarchically structured dictionaries to efficiently search for sparse linear combinations of exemplars.

5. GENERAL DISCUSSION AND CONCLUSIONS

In this paper, we investigate several aspects of Sparse Classification (SC) that are of interest when developing a practical implementation of SC. First, we addressed the computational complexity of SC and showed that large speedups can be obtained with a mainstream GPU and with only a minimum of effort. Recognition experiments show that the use of single precision floating point variables necessary for obtaining these speedups does not result in a significant decrease in accuracy. With speedups of up to 28, consistent with those reported on in literature for similar applications [4], real-time processing using SC seems now within our grasp.

A second issue that was addressed is the noise robustness of SC in the presence of noise types not present in the noise dictionaries. By combining noise exemplars extracted from a noise database with artificial noise exemplars, good noise robustness is obtained on both matched and mismatched noisy test sets. In a third experiment, the influence of enforcing sparsity was investigated. It was found that for both the speech and noise parts of the dictionary, the use of sparsity is beneficial. At the same time, it became clear that the sparsity weights were SNR-dependent and should be governed by hyper-parameters.

Finally, the influence of speech dictionary size was investigated. It was found that the larger the speech dictionary, the better the recognition accuracy on both clean and noisy speech, although the returns are diminishing. With the issue of computational complexity largely resolved, research on SC should now focus on more robust and adaptive methods to determine the necessary tuning parameters, efficient ways of using larger dictionaries, and investigate the effectiveness of SC on larger vocabulary tasks.

Acknowledgements

The research of Jort F. Gemmeke was supported by the Dutch-Flemish STEVIN project MIDAS and by IWT project ALADIN. The research of Tuomas Virtanen and Antti Hurmalainen has been funded by the Academy of Finland. The research of Yang Sun has received funding from European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 213850.

REFERENCES

[1] L. Deng and H. Strik, "Structure-based and template-based automatic speech recognition - comparing parametric and non-parametric approaches," in *Proc. INTERSPEECH*, 2007, pp. 898–901.

[2] Jort F. Gemmeke and Tuomas Virtanen, "Noise robust exemplar-based connected digit recognition," in *Proc. ICASSP*, 2010.

[3] Tuomas Virtanen, Jort F. Gemmeke, and Antti Hurmalainen, "State-based labelling for a sparse representation of speech and its application to robust speech recognition," in *Proc. Interspeech*, 2010.

[4] Pradeep Nagesh, Rahul Gowda, and Baoxin Li, "Fast GPU implementation of large scale dictionary and sparse representation based vision problems," in *Proc. ICASSP*, 2010, pp. 1570–1573.

[5] Jort F. Gemmeke and Tuomas Virtanen, "Artificial and online acquired noise dictionaries for noise robust ASR," in *Proc. Interspeech*, 2010.

[6] "GPUmat: GPU toolbox for MATLAB," *Online: http://gp-you.org/*, 2010.

[7] Jort F. Gemmeke, L. ten Bosch, L.Boves, and B. Cranen, "Using sparse representations for exemplar based continuous digit recognition," in *Proc. EUSIPCO*, Glasgow, Scotland, August 24–28 2009, pp. 1755–1759.