

ROBUSTNESS OF PHONEME RECOGNITION USING SUPPORT VECTOR MACHINES

Lena Khoo¹, Zoran Cvetković², Peter Sollich¹

King's College London

¹Department of Mathematics and ²Division of Engineering

Strand, London, WC2R 2LS, UK

{zoran.cvetkovic,peter.sollich}@kcl.ac.uk

ABSTRACT

The robustness of phoneme recognition using support vector machines to additive noise is investigated for three kinds of speech representation. The representations considered are PLP, PLP with RASTA processing, and a high-dimensional principal component approximation of acoustic waveforms. While the classification in the PLP and PLP/RASTA domains attains superb accuracy on clean data, the classification in the high-dimensional space proves to be much more robust to additive noise.

1. INTRODUCTION

Substantial research efforts over the past decades, devoted to the higher levels of speech recognition systems, *i.e.* language and context modeling, have resulted in major breakthroughs that have made automatic speech recognition (ASR) possible. ASR systems, however, still lack the level of robustness inherent to human speech recognition [6, 9]. While language and context modeling are essential, reducing many errors in speech recognition, the importance of robust recognition of nonsense syllables does not appear to have been sufficiently appreciated, despite the fact that it is well known that humans attain a major portion of their inherent robustness in speech recognition early on in the process, before and independently of context effects [3, 7, 8]. Language can be thought of as a layer of redundancy that is built into speech, and while exploiting this by language modeling is clearly important, it is not sufficient for making automatic speech recognition work optimally. From the information theoretic perspective, language and context are channel codes and they can successfully decode messages carried by speech signals only if the elementary speech units which are fed into them are recognized sufficiently accurately; in the extreme case when phonemes or syllables are recognized at the level of chance (random guessing), no context and language modeling can retrieve any information from speech. In recognizing syllables or isolated words, the human auditory system performs above chance level already at -18dB SNR (signal-to-noise ratio) and significantly above it at -9dB SNR [3, 7, 8]. No automatic speech classifier is able to achieve performance close to that of the human auditory systems in recognizing such isolated words or phonemes under severe noise conditions, as has been reconfirmed recently in an extensive study by Sroka and Braida [9].

The first step in all speech recognition algorithms is to represent consecutive speech segments using low-dimensional feature vectors. Two major reasons for using feature vectors are to represent speech in a low-dimensional space in order to facilitate accurate estimation of probability density functions from limited data,

and to remove non-lexical variability irrelevant to recognition, *e.g.* speaker related nuances such as pitch, time alignment, etc. The accuracy of ASR continues to improve, with many new judicious approaches to feature extraction, but the problem of robustness persists.

Speech production is a channel coding process designed to embed redundancy in speech waveforms in a highly structured manner, keeping different speech units far apart from each other so that they can withstand a significant amount of additive noise and mangling distortion before they overlap significantly. Any state-of-the-art ASR front end performs a considerable compression of speech signals, representing them in a space of a relatively low dimension where different speech units, even though separated, may not be sufficiently apart from each other; they may then overlap considerably already at lower noise levels than in the original domain of acoustic waveforms. We are not of course arguing that speech units such as phonemes, syllables or sub-phonetic units do not overlap in the acoustic waveform domain; they certainly do. However, it is quite possible that in the presence of noise the overlap is smaller in the space of acoustic waveforms, or a high-dimensional approximation of it, than in low-dimensional spaces of feature vectors. In addition, nonlinear transformations which take place in feature extraction algorithms may complicate the structure of the sets corresponding to different speech units, so that more sophisticated classification procedures may be required than for classification in the acoustic waveform domain or in some high-dimensional linear approximation space.

To test this hypothesis about separation of phonetic units in different representation domains and the impact of dimension reduction we perform classification of phonemes using support vector machines (SVMs) for three different representations: PLP [4], PLP with RASTA processing [5], and a high-dimensional principal component (PC) representation of acoustic waveforms of speech. The details of the classification techniques are provided in Section 2. The experiments, the results of which are reported in Section 3, show that while the classification using PLP and PLP/RASTA representations achieves considerably better results on clean data than the classification using the PC representation, it is much more sensitive to additive noise.

2. PHONEME CLASSIFICATION USING SUPPORT VECTOR MACHINES

The data set used in this study are 64ms segments (1024 samples) of phonemes from the TIMIT data base. For the purpose of our proof-of-concept study, the classification task was restricted

to distinguishing the following six phonemes: /b/, /t/, /m/, /r/, /l/, and /z/. This still requires a multiclass classification algorithm. We used SVMs as base classifiers for distinguishing two groups of phonemes at a time; a number of these binary classifiers are then combined using error-correcting code (ECC) methods. In order to separate K ($= 6$, in our case) classes, one proceeds as follows [1]. A total of B binary classifiers are trained to distinguish between specific subsets of phonemes. The allocation of these subsets is determined by a $K \times B$ matrix \mathbf{M} with elements $M_{kb} \in \{0, \pm 1\}$: classifier b receives as training data the phonemes with class labels k for which $M_{kb} \neq 0$, with effective class labels $+1$ or -1 as determined by M_{kb} . It therefore learns to separate phonemes with $M_{kb} = 1$ from those with $M_{kb} = -1$, but has no knowledge about phoneme classes k with $M_{kb} = 0$. In the simplest case of one-vs-all classification, $B = K$ and the b -th column of \mathbf{M} has a $+1$ entry in row b and -1 's elsewhere; classifier b then learns to distinguish phoneme class b from all others. In the opposite extreme of pairwise classification, each classifier sees only two phoneme classes. Then $B = K(K - 1)/2$, and the elements of \mathbf{M} are zero except for one $+1$ and one -1 in each column. We also investigated other choices: (1) Three-vs-three, where a total of $B = 20$ classifiers are trained on all possible splits of the $K = 6$ classes into two groups of three; here each column of \mathbf{M} contains exactly three $+1$'s and three -1 's. (2) Random dense \mathbf{M} with $B = 25$, where each element $M_{kb} = \pm 1$ with equal probability. We sampled 10,000 such matrices, excluded those which had any columns or rows with all elements equal, and chose the one with the maximum smallest Hamming distance between columns (or their negatives). (3) A random sparse ECC matrix \mathbf{M} was produced in the same way, except that $M_{kb} = 0$ with probability 0.5 and $= \pm 1$ with probability 0.25 each.

To make predictions with the resulting array of binary classifiers, one obtains for a given input \mathbf{x} the outputs (decision values) $f_b(\mathbf{x})$ of all B classifiers. If \mathbf{x} is in class k , one expects these outputs to follow the pattern set by the k -th row of the matrix \mathbf{M} . The predicted class $F(\mathbf{x})$ is therefore chosen as the one for which the vector $\{M_{kb}\}$ ($b = 1 \dots B$) is closest to $\{f_b(\mathbf{x})\}$:

$$F(\mathbf{x}) = \arg \min_k d(\{M_{kb}\}, \{f_b(\mathbf{x})\}) \quad (1)$$

The relevant distance measure is taken to be of the form

$$d(\{M_{kb}\}, \{f_b(\mathbf{x})\}) = \sum_{b=1}^B L(z_{kb}) \quad (2)$$

where L is some loss function and $z_{kb} = M_{kb} f_b(\mathbf{x})$ defines a *margin*: classifier b predicts that \mathbf{x} belongs to one of the phoneme classes k with $z_{kb} > 0$, and does not belong to one of the classes with $z_{kb} < 0$; for classes with $z_{kb} = 0$ it makes no definite prediction (since it not been trained on them). We experiment with three different choices of the loss function:

- Hamming loss: the hard predictions $\text{sgn}(f_b(\mathbf{x}))$ are used instead of the decision values $f_b(\mathbf{x})$ themselves, and a loss of 0 or 1 assigned depending on whether or not the former agree with M_{kb} . This corresponds to $L(z) = [1 - \text{sgn}(z)]/2$; for unseen classes this loss function contributes $1/2$ to the distance d . In pairwise classification, minimizing the resulting distance gives majority voting, where the predicted class is the one having the most "for" votes among the classifiers that were trained on it.
- Hinge loss: $L(z) = (1 - z)_+ = \max\{z, 0\}$.

- Exponential loss: $L(z) = e^{-z}$.

For one-vs-all classification, the distance (2) is easily seen to be $L(f_k(\mathbf{x})) - L(-f_k(\mathbf{x}))$ up to a k -independent constant, so that the last two choices correspond to max-wins, i.e. predicting the class k which has the largest among the $B = K$ decision values f_b .

As regards the binary SVM classifiers, preliminary tests showed that linear kernels resulting in unsatisfactory performance; we therefore used the more general radial basis function kernel [2]. The two resulting SVM parameters (kernel width σ and misclassification penalty C) were tuned individually for each classifier by cross-validation. Each classifier was trained on 800 examples from each phoneme class included in its training set, and we tested the overall multiclass predictions on 200 test examples per phoneme. In the one-vs-all case, somewhat better performance can be obtained by balancing each classifier's training set (800 examples from the $+1$ -phoneme, 160 examples each from the five other phonemes; results not shown).

3. CLASSIFICATION RESULTS

Classification of the six selected phonemes is performed on 64ms (1024-sample) segments of their acoustic waveforms, using PLP and PLP/RASTA representations, and a high-dimensional PC representation of these acoustic waveforms. The PC representation is generated by projecting acoustic waveforms onto the first 600 principal components obtained by performing principal component analysis (PCA) of the set of all available waveforms. ECC multiclass classifiers are built using the following coding matrices: one-versus-all, pairwise, three-versus-three, random dense, and random sparse; and each of these is combined with one of the three loss functions: Hamming, hinge, and exponential, as described in the previous section. This gives 15 classification methods. The SVM classifiers are trained and tuned on clean data, and then the classification was performed on clean data, and noisy data with noise levels of 12, 6, 0 and -6 dB SNR.

The results of the classification using the PCA data are shown in Figure 1. The misclassification (test error) rate for most of the 15 methods is around 20% for clean data. At -6 dB SNR most of the classifiers were operating at chance level, with error rates near $5/6=83\%$, and the best result, 72% error rate was obtained with pairwise classification and hinge cost function. One observes that additive noise does not considerably affect misclassification rate up to 0dB SNR.

The results of the classification in the PLP and PLP/RASTA domains are displayed in Figure 2 and Figure 3. One can see that both PLP and PLP/RASTA give excellent results for most of the classifiers when tested on clean data. This again confirms the effectiveness of these two representations in capturing that part of the information in the acoustic waveforms which is most relevant for recognition. However, the performance of almost all classifiers degrades considerably with even small levels of noise. This performance degradation is particularly drastic in the case of the PLP/RASTA representation where almost all the classifiers tested operate at chance level already at 12dB SNR. On the other hand, classification in the PLP and PLP/RASTA domains gave very good results when classifiers were trained on noisy data and tested on data with the same noise level. In particular, some classifiers attained classification errors around 20% at -6 dB SNR if the classifiers were trained under the same noise conditions.

It is worth noting that when going from acoustic waveforms to the PLP representation the dimension of the representation space is reduced by a factor around twenty. The loss of robustness in classification in the case when there is a mismatch between training and testing conditions may be a result of this drastic dimension reduction. Note also that the classifiers were trained on the same amount of data for all three representations. Conclusions about the superiority of classification in the PLP and PLP/RASTA domains compared to the classification in the PCA domain therefore require some caution: relative to the dimension of the representation space, the classifiers in the PLP and PLP/RASTA domains were trained on around ten times larger data sets.

4. CONCLUSION

We have presented results of phoneme classification using SVMs with three representations of speech signals: PLP, PLP/RASTA, and a high-dimensional PC representation. Fifteen different SVM-based multiclass classification algorithms were investigated. The results show that while the PLP and PLP/RASTA representations facilitate very accurate classification of clean data, their performance degrades significantly with even small levels of additive noise on the test data. On the other hand, the classification using the high-dimensional PC representation, even though inferior on clean data, gives better results when speech is degraded by additive noise.

Acknowledgment:

Zoran Cvetković is very grateful to Jont Allen and Bishnu Atal for their support and guidance in this research.

5. REFERENCES

- [1] E. Allwein, R. E. Schapire, and Y. Singer, "Reducing multi-class to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, **1**:113–141, 2001.
- [2] N. Cristianini and J. Shawe-Taylor: *An introduction to Support Vector Machines*, Cambridge University Press, 2000.
- [3] N. R. French and J. C. Steinberg, "Factors governing the intelligibility of speech sounds," *J. Acoust. Soc. Am.* **19**:90–119, 1947.
- [4] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Am.* **87**:1738–1752, 1990.
- [5] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Processing*, **2**:578–589, 1994.
- [6] R. P. Lippmann, "Speech Recognition by Humans and Machines," *Speech Communication*, **22**: 1–15, 1997.
- [7] G. A. Miller, A. G. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of the test material," *J. Exp. Psychol.* **41**:329–335, 1951.
- [8] G. A. Miller and P. E. Nicely, "An analysis of perceptual confusions among some English consonants," *J. Acoust. Soc. Am.* **27**: 338–352, 1955.
- [9] J. J. Sroka and L. D. Braida, "Human and machine consonant recognition," *Speech Communication*, **45**:401–423, 2005.

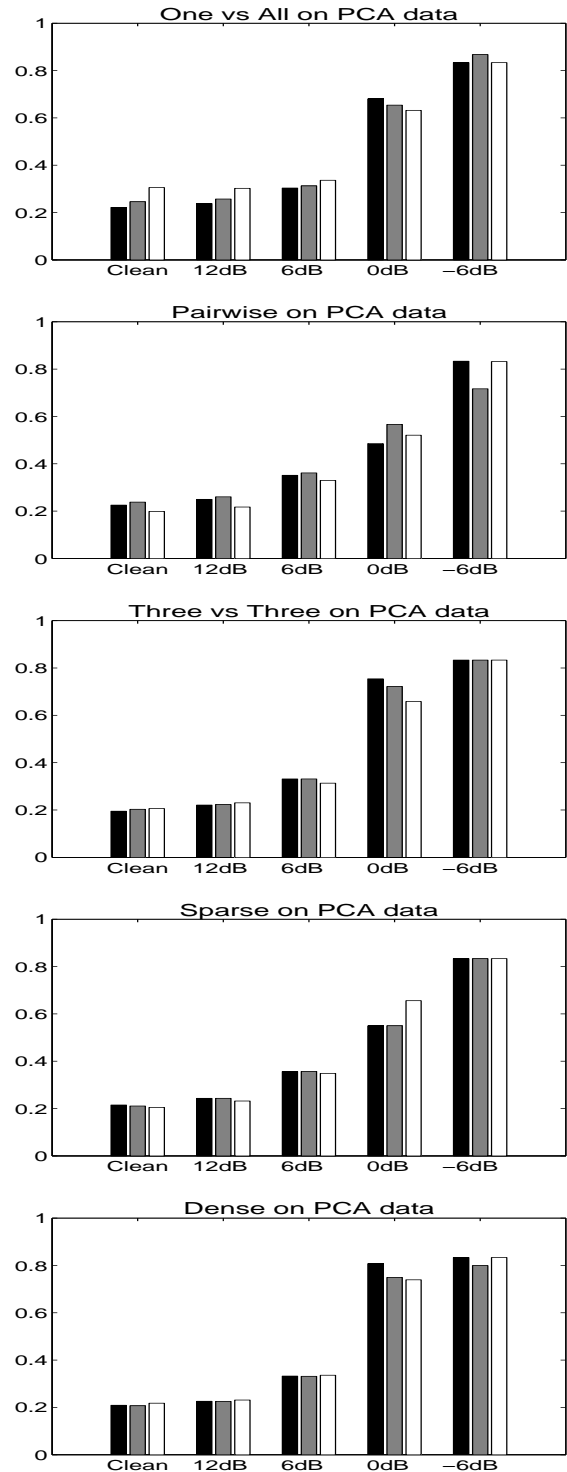


Figure 1: Classification in the PCA domain. Test error (misclassification) rates are for different ECOC matrices M as indicated above each graph. The SNR increases from left to right in both columns; within each group of three results the loss function L is exponential, hinge and Hamming from left to right.

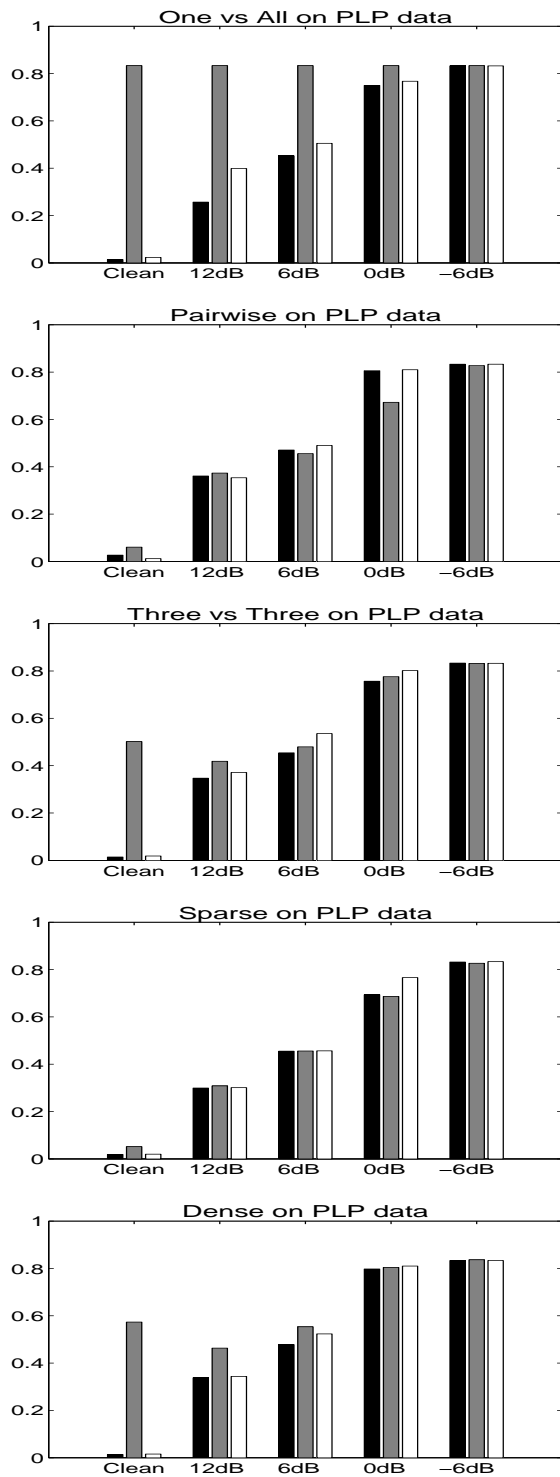


Figure 2: Classification in the PLP domain. Test error (misclassification) rates are for different ECOC matrices \mathbf{M} as indicated above each graph. The SNR increases from left to right in both columns; within each group of three results the loss function L is exponential, hinge and Hamming from left to right.

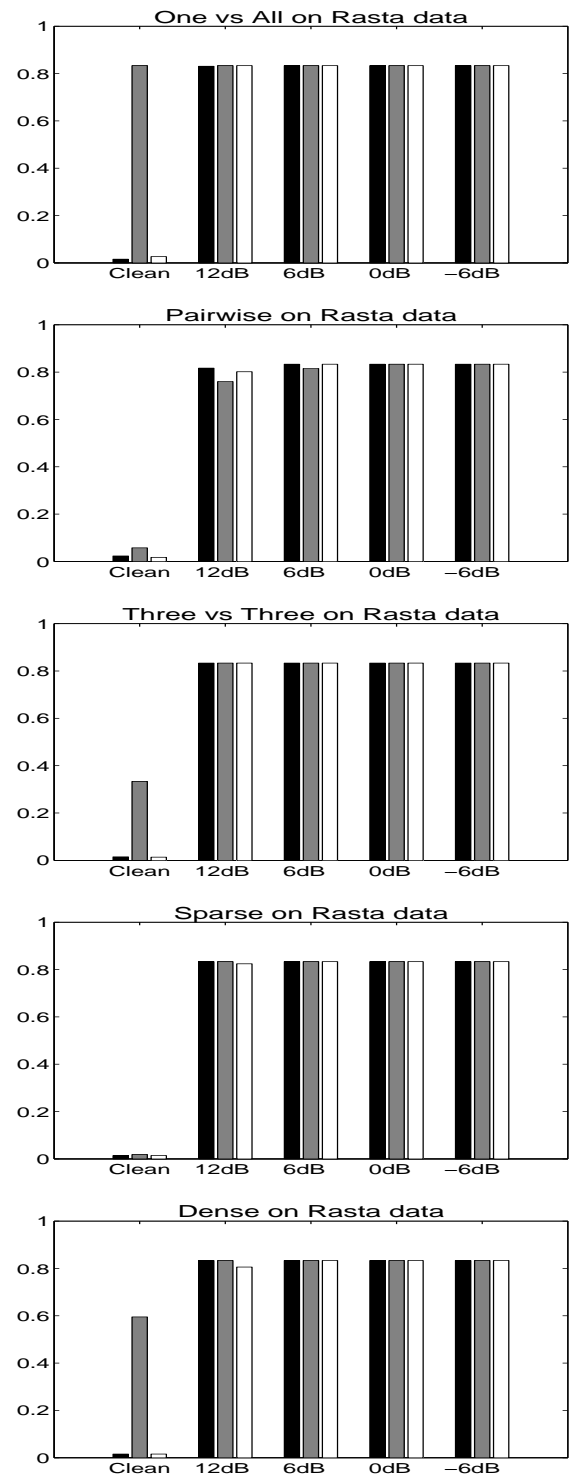


Figure 3: Classification in the RASTA domain. Test error (misclassification) rates are for different ECOC matrices \mathbf{M} as indicated above each graph. The SNR increases from left to right in both columns; within each group of three results the loss function L is exponential, hinge and Hamming from left to right.