Speech Recognition With State-based Nearest Neighbour Classifiers

Thomas Deselaers, Georg Heigold, and Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department – RWTH Aachen University, Aachen, Germany
{deselaers,heigold,ney}@cs.rwth-aachen.de

Abstract

We present a system that uses nearest neighbour classification on the state level of the hidden Markov model. Common speech recognition systems nowadays use Gaussian mixtures with a very high number of densities. We propose to carry this idea to the extreme, such that each observation is a prototype of its own. This approach is well-known and widely used in other areas of pattern recognition and has some immediate advantages over other classification approaches, but has never been applied to speech recognition. We evaluate the proposed method on the *SieTill* corpus of continuous digit strings and on the large vocabulary EPPS English task. It is shown that nearest neighbour outperforms conventional systems when training data is sparse.
Index Terms: automatic speech recognition, nearest neighbour classification, kernel densities

1. Introduction

In common speech recognition systems, Gaussian mixture models (GMMs) are applied at the state level of the hidden Markov Model. Recently, the number of densities used is increased to optimise recognition performance. Carrying this trend to the extremes would result in a Gaussian mixture model with very few observations per density. Carrying this even further results in nearest neighbour classification where each training observation is employed as a prototype.

Nearest neighbour classification is a widely adopted technique in many pattern recognition applications e.g. in image recognition [1] and protein identification [2]. The nearest neighbour method is well understood and offers some immediate advantages: on the theoretical side, it can be proven that the nearest neighbour classifier performs asymptotically optimal if very little or very much training data is available [3]. On the practical side, the nearest neighbour is easily implemented and training is very easy. The training phase of the algorithm only consists of storing the feature vectors and class labels of the training samples. Furthermore, when setting up a nearest neighbour classifier very few parameters have to be tuned compared to other classification methods where several parameters have to be set.

The use of the nearest neighbour technique in speech recognition is problematic for the following reasons: (i) The amount of training data commonly used in speech recognition poses a problem because for an efficient nearest neighbour classification it is essential to keep all data in RAM. (ii) The comparison of an observation to all training samples obviously requires more computation time than the comparison with a set of Gaussian densities which typically consists of several orders of magnitude less densities than there are training observations. (iii) In speech recognition commonly hidden Markov models (HMMs) are used which require probability estimates at the state level. The nearest neighbour classifier does not directly allow for the estimation of probabilities which makes the integration into the HMM problematic. The kernel densities [4] approach, is a common extension to the nearest neighbour method which allows to determine probability estimates.

Related Work. To our knowledge, nearest neighbour classification has never been used at the state level in a speech recognition system. In [5] the author proposes a nearest neighbour classifier at word level in a system for small vocabulary gesture recognition with whole-word models: a sequence to be recognised is aligned to every training sequence using the conventional dynamic programming time alignment algorithm, then a distance is calculated between the two sequences and the class of the training observation with the lowest distance is chosen. In [6] approximation techniques from nearest neighbour classification are used to improve the efficiency of score calculation in GMM-based HMM. In [7] nearest neighbour classification is used for speech channel segmentation in a co-channel environment.

In the image recognition domain, a combination of many nearest neighbour decisions has been used for the classification of faces [8] and in most content-based image retrieval systems nearest neighbour techniques are applied to find similar images for a given query[9].

In this work, we propose to use a nearest neighbour classifier at the state level in a speech recognition system. This is made possible by the use of efficient approximate nearest neighbour search using kd-trees [10].

2. Nearest Neighbour Classification

A common speech recognition system uses Bayes' decision rule. Given a sequence of feature vectors x_1^T, the sequence of words $\hat{w}_1^N$ is obtained as

$$\begin{aligned} \hat{w}_1^N &= \arg\max_{w_1^N} \left\{ p(w_1^N|x_1^T) \right\} && (1) \\ &= \arg\max_{w_1^N} \left\{ p(w_1^N) \cdot p(x_1^T|w_1^N) \right\} && (2) \\ &= \arg\max_{w_1^N} \left\{ p(w_1^N) \cdot \sum_{s_1^T} p(x_1^T, s_1^T|w_1^N) \right\}. && (3) \end{aligned}$$

Then, $p(x_1^T, s_1^T|w_1^N)$ is further reformulated using Markov assumption as

$$\begin{aligned} p(x_1^T, s_1^T|w_1^N) &= \prod_{t=1}^{T} p(x_t, s_t|x_1^{t-1}, s_1^{t-1}, w_1^N) && (4) \\ &= \prod_{t=1}^{T} p(x_t|s_t) \cdot p(s_t|s_{t-1}, w_1^N). && (5) \end{aligned}$$

$p(x_t|s_t)$ is commonly modelled as a mixture of Gaussian densities using up to 500 densities per state summing up to a total of about 1,000,000 densities [11]. It can be observed that using even more densities leads to better results. Here we replace it by a nearest neighbour classifier where effectively each training observation is a density of its own.

The decision rule of the nearest neighbour classifier for a single observation vector x is

$$k = \arg\min_k \left\{ \min_{n=1,\ldots,N_k} ||x - x_{nk}||^2 \right\} \quad (6)$$

where N_k is the number of training observations from class k and x_{nk} is the n-th observation from this class.

To integrate the nearest neighbour classifier into the HMM, it is necessary to obtain probabilities $p(x_t|s_t)$ for each state s_t. We model these probabilities as

$$p(x|k) \propto \exp\left(- \min_{n=1,\ldots,N_k} ||x - x_{nk}||^2\right) \quad (7)$$

where the different k represent the different possible states s_t.

An important feature of a speech recognition system is efficiency because huge amounts of data need to be handled. Exhaustive search of the complete set of training data for a nearest neighbour of an observation is prohibitively expensive. Therefore, we use a kd-tree for efficient approximate search of nearest neighbours[10]. We only give a short outline of the method:

In a kd-tree, the search for the nearest neighbour of each observation is done recursively and starts from the root of the tree. The root represents the whole feature space. At each node the feature space is subdivided and the search continues in the part of the tree containing the test observation. Once a leaf-node is reached, all prototypes contained in that leaf are compared to the test observation. This is not a complete process, i.e. it might happen that the nearest neighbour candidate is not the actual nearest neighbour, because it may happen that at some stage, the subtree chosen is not the one containing the real nearest neighbour of the observation. However, it is possible to determine all subtrees that may still contain prototypes closer to the observation than the current nearest prototype. Using backtracking, the search can be expanded such that exactness can be guaranteed.

In our setup, an exact solution might not be necessary and in this case, the backtracking process can be aborted as soon as a certain exactness criterion is reached. Here, the criterion is that it can be guaranteed that no nearest neighbour with a distance smaller than the distance of the currently determined prototype minus some preset value ε can exist. This concept allows for efficient search of nearest neighbours among very large sets of prototype vectors.

2.1. k Nearest Neighbours

A common extension to the nearest neighbour approach to reduce the influence of noise in the training data is to use not just one nearest neighbour but a set of k nearest neighbours. Then a voting scheme (e.g. majority voting) is used to make a decision. It is not straightforward to incorporate this concept into our system because here the neighbours are sought class (i.e. state) wise and interaction between the states would incur a significantly higher computational cost. It is possible to use k training observations per state and average the distances which directly leads to the *kernel densities* method described in the next paragraph.

2.2. Kernel Densities

The kernel densities approach is an extension to the nearest neighbour approach that allows for the estimate of emission probabilities $p(x|k)$ [4]. These probabilities are obtained as

$$p(x|k) = \frac{1}{N_k} \sum_{n=1}^{N_k} \exp\left(-\frac{1}{2}\frac{||x - x_{nk}||}{\sigma}\right) \quad (8)$$

where σ is a scaling factor. The emission probabilities $p(x|k)$ are calculated class wise and thus, this approach fits perfectly into our system. The efficiency of this classifier can be improved by approximating the sum by using the subset of the M best matches (cf. k nearest neighbours). This approximation is commonly applied and it is known hat it hardly has an impact on the classification results, because of the exponential decrease of influence in the distance between two vectors. Note that the nearest neighbour approach presented above is effectively doing exactly this with a subset of size 1.

2.3. Reducing the Amount of Silence Observations

In most corpora for automatic speech recognition, *silence* and different types of *noise* have a large share of the data and are modelled jointly. Therefore the states modelling silence and noise have an enormous amount of training data at their disposal. Although normally having a large amount of training data is a good thing to learn better models, in some cases, like here, it is not an advantage. Contrary to a conventional GMM-based system where it is guaranteed that the sum of the cluster weights for each mixture is normalised to one, in the nearest neighbour approach the number of observations per state strongly influences the implicit prior probability of that particular state. On the one hand, having very many observations in one state makes the nearest neighbour search inefficient. On the other hand the high amount of noise among the silence data is likely to be confused with proper utterances of words [12]. The reduction of the silence observations corresponds to implicitly reducing the prior probability of silence.

To avoid these problems, we apply two different methods to reduce the amount of silence used in the models:

Energy-dependent Silence Reduction: To reduce the amount of noise among the silence observations, all silence observations with energy above a certain threshold are discarded.

Randomised Silence Reduction: A randomly selected portion of all silence observations is discarded.

2.4. Scaling of Feature Vectors

For efficiency reasons we use the Euclidean distance in the nearest neighbour classification. Therefore the different components of the feature vectors might have a different impact on the obtained distances due to their non-uniform variances.

To address this issue we calculate the pooled leaving-one-out covariance matrix Λ^2 (i.e. the second centered moment of each training observation to its nearest neighbor training observation) and transform the vectors x by

$$x' = \Lambda^{-1}x. \quad (9)$$

The matrix Λ^2 is obtained as

$$\Lambda^2 = \sum_x (x - \hat{x})(x - \hat{x})^T \quad (10)$$

where $\hat{x}$ is the nearest neighbour of x from the same class.

3. Experiments and Results

In the following sections, we describe experiments that were performed on the *SieTill* corpus [13] of telephone line recorded German continuous digit strings. On this corpus we tune the performance of the nearest neighbour based method and compare it with a GMM-based system. Then we describe experiments with varying amounts of training data on the English

EPPS large vocabulary task and compare the performance with a GMM-based system.

The *SieTill* corpus consists of approximately 43,000 spoken digits in 13,000 sentences for training and test set. The recognition system is setup gender-dependent using whole-word HMMs. For each gender 214 distinct states and one silence state are used. Feature vectors used are MFCC features with a temporal window of length 5 which are LDA transformed keeping 25 components.

The EPPS English task contains recordings from the European Parliament Plenary Sessions (EPPS). 87.5h of speech recordings/704,883 running words were manually transcribed, which are used for training of the acoustic models [11]. The non-speech proportion is roughly 30%. MFCC and a voicing feature are used as acoustic features, 9 consecutive frames are concatenated and LDA-reduced to 45 dimensions. The MFCC features are warped using a fast variant of vocal tract length normalisation. The triphones are clustered using CART, resulting in 4,501 generalised triphone states. The acoustic models are trained on the complete manually transcribed data. The development data from the evaluation campaign 2006 comprise 3.2h/27,029 running words. For recognition the vocabulary size is 52,429 and a 4-gram language model is used.

3.1. Baseline experiments

Using the default speech recognition system with nearest neighbour classifiers instead of normal Gaussian mixtures at the states results after tuning of the standard parameters in a word error rate of 2.7% with about 7 times as many deletions as insertions (cf. Table 1, line 'nearest neighbour baseline'). Decreasing the word-penalty could not reduce the number of deletions but rather increased the number of substitution errors, only. The high number of deletions is probably due to the large variability in the silence observations which are easily confused with utterances of digits in the test phase.

3.2. Reduction of Silence Observations

To reduce the variability of the training data of the silence state, the amount of training data for this particular state is reduced as explained in Section 2.3. First we discard all silence observations with energy above a certain threshold which should correspond to those silence observations that contain a high amount of noise and are therefore likely to be confused with words. We experimented with various thresholds and the results are shown in Figure 1. On the x-axis, the total number of silence observations is given and on the y-axis the number of deletions, insertions, and the WER are given in %. The results show that with decreasing number of observations the WER and the number of insertions increases much quicker than the number of deletions decreases and that therefore no improvement has been achieved. This effect can be explained by the noise in the test data which is now confused with proper utterances of words because it cannot be explained by the silence model anymore.

In Figure 1 also the results of experiments where the number of silence observations was randomly reduced are given. Here, an improvement is obtained when only 10% to 20% of the observation vectors are used. The reduction of the deletion errors is stronger than the enlargement of the insertion errors and at the same time the number of substitutions is slightly reduced. An improved word error rate of 2.19% with 0.45% and 0.27% deletions and insertions respectively is achieved. Therefore we stick with using approximately every sixth silence observation for the forthcoming experiments.

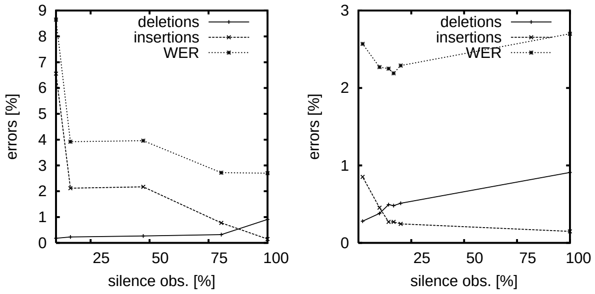

Figure 1: Errors depending on the reduction of silence observations. left: energy dependent, right: random.

3.3. k-Nearest Neighbours and Kernel Densities

Figure 2 shows the results of experiments using the k-nearest neighbour technique as described in Section 2.1. Using more than one neighbour from each class does not lead to an improvement of the results but rather increases the WER. Note, that the number of deletion, insertion and substitution errors increases at approximately the same rate.

Using the kernel density approach leads to an improved performance as can be seen from Table 1 line '*kernel densities*'. Here, the combination of multiple prototypes per class leads to an improved robustness of the method.

3.4. Scaling of Feature Vectors

The different components of the feature vectors after LDA are of different importance and in different value ranges. To account for this observation, we transform all feature vectors according to equation 9. The scaling hardly effects the results and therefore has not been followed further.

3.5. Efficiency

As described above, the efficiency of the nearest neighbour search depends on the required precision. In Figure 3, the dependence of the time for recognition depending on the accuracy requirement of the kd-tree search is given. The experiments were performed on an 1.86GHz machine with 2GB RAM. The recognition accuracy is hardly affected by this parameter and the real-time factor of the system with ε=50 which corresponds to using no backtracking at all is 0.58. Counterintuitively, the best recognition accuracy is obtained using a high ε which is in accordance to results reported in [14]. This effect can be explained as the result of smoothing. Using a high tolerance in the nearest neighbour search effectively corresponds to a smoothing of the training data. For comparison, the real-time factor of a system using Gaussian mixture with 64 densities per state has a real-time factor of 0.15 which can be improved to 0.05 using vectorisation and single-instruction multiple data instructions in modern computers [15].

3.6. Amount of Training Data

As described above, it can be shown that nearest neighbour classification performs asymptotically optimal if very small or very large amounts of training data are available. In this section we compare the nearest neighbour system with a GMM-based section using varying amounts of training data on the development corpus of the EPPS English task. Results for different amounts of training data are given in Figure 4. It can be seen that when using 9h of training data the GMM-system outperforms the nearest neighbour system. With smaller amounts of available training data, as expected, the performance of both systems deteriorates but the nearest neighbour system is more

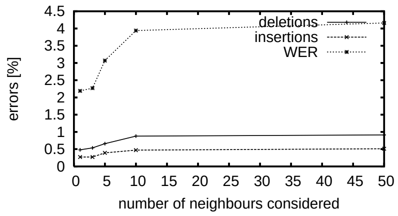

Figure 2: Error rates depending on the number of neighbours used in the recognition.

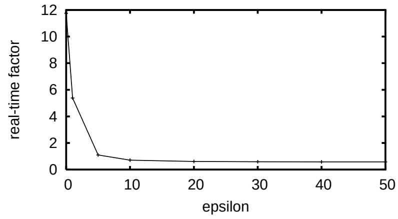

Figure 3: Real-time factors depending on the precision ε of the nearest neighbour search.

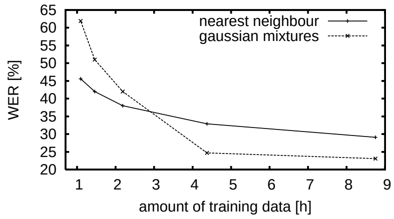

Figure 4: WER depending on the amount of available training data for the EPPS English task.

Table 1: Summary of the results obtained in the experiments compared to a speech recognition system using GMMs on the *SieTill* corpus.

method	del [%]	ins [%]	WER [%]
GMM [13]	0.46	0.38	1.84
nearest neighbour baseline	0.90	0.14	2.70
nearest neighbour (silence red.)	0.45	0.27	2.19
kernel densities	0.37	0.28	1.96

robust w.r.t. the lack of training data. When only 3 hours or less of training data are available, the nearest neighbour based system outperforms the GMM-based system. Note, that for the experiments with GMMs we tested models with 2 to 64 densities and always chose the system that performed best to account for the different amounts of training data available.

4. Discussion

Table 1 gives an overview on the results obtained on the *SieTill* corpus. Starting from a baseline of 2.7% WER, by gradually addressing the problems in the recognition, finally a WER of 1.96% is obtained using a reduced number of silence observations and kernel densities. This result is comparable with results obtained from GMM-based conventional speech recognition system using approximately 14,000 densities.

The results on EPPS corpus show that the nearest neighbour approach can benefit from its theoretical advantage when only sparse training data is available which might help e.g. for languages where only few hours of training data are available. and where it is not possible to reliably estimate a GMM.

5. Conclusions & Outlook

We presented a method using nearest neighbour classification techniques at the state level of a speech recognition system which has, to our knowledge, not been presented before. The results obtained on the *SieTill* corpus and on the EPPS corpus are promising and a clear advantage of the nearest neighbour method in the case where only small amounts of training data are available was shown which is consistent with theory.

This effect allows to hope that the nearest neighbour based speech recognition might be applicable for speech recognition of languages with very sparse training data. Furthermore, the nearest neighbour based system might be an interesting approach to use in model combination under certain conditions.

6. Acknowledgements

We would like to thank P. Dreuw, C. Gollan, S. Hahn, B. Hoffmeister, and R. Schlüter for discussing the method and J. Cano from ITI Valencia for his kd-tree library.

This material is partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-06-C-0023, and was partly funded by the European Union under the integrated project TC-STAR (FP6-506738) and by the DFG (Deutsche Forschungsgemeinschaft) under contract NE-572-6. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA.

7. References

[1] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, "Deformation models for image recognition," *IEEE Trans. Pattern Recognition and Machine Intelligence*, to appear, 2007.

[2] M. Ankerst, G. Kastenmüller, H.P. Kriegel, and T. Seidl, "Nearest neighbor classification in 3d protein databases," in *Int. Conf. on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, aug 1999, pp. 34–43.

[3] G. Loizou and S.J. Maybank, "The nearest neighbor and the Bayes error rates," *IEEE Trans. Pattern Recognition and Machine Intelligence*, vol. 9, no. 2, pp. 254–262, mar 1987.

[4] B.W. Silverman, *"Density Estimation for Statistics and Data Analysis"*, Chapman and Hall, New York, 1986.

[5] P. Dreuw, "Appearance-based gesture recognition," Diploma thesis, RWTH Aachen University, Aachen, Germany, Jan. 2005.

[6] B.L. Pellom, R. Sarikaya, and J.H.L. Hansen, "Fast likelihood computation techniques in nearest-neighbor based search for continuous speech recognition," *IEEE Signal Processing Letters*, vol. 8, no. 8, Aug. 2001.

[7] J.K. Shah, B.Y. Smolenski, R.E. Yantorno, and A.N. Iyer, "Sequential k-nearest neighbor pattern recognition for usable speech classification," in *European Signal Processing Coference*, Vienna, Austria, Sept. 2004.

[8] R. Paredes, J. Perez-Cortes, A. Juan, and E. Vidal., "Local representations and a direct voting scheme for face recognition," in *Workshop on Pattern Recognition in Information Systems*, Setúbal, Portugal, July 2001, pp. 71–79.

[9] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[10] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silvermann, and A.Y. Wu, "An optimal algorithm for approximate nearest neighbor searching," *Journal of the ACM*, vol. 45, no. 6, pp. 101–124, Nov. 1998.

[11] J. Lööf, M. Bisani, Ch. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. Schlüter, and H. Ney, "The 2006 RWTH parliamentary speeches transcription system," in *Int. Conf. Spoken Language Processing*, Pittsburgh, PA, Sept. 2006, pp. 105–108.

[12] A. Ljolje, "Optimization of class weights for LDA feature transformations," in *Int. Conf. on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, USA, Sept. 2006, pp. 385–388.

[13] R. Schlüter, W. Macherey, B. Müller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34, pp. 287–310, May 2001.

[14] T. Kölsch, D. Keysers, H. Ney, and R. Paredes, "Enhancements for local feature based image classification," in *International Conference on Pattern Recognition*, Cambridge, UK, Aug. 2004, vol. 1, pp. 248–251.

[15] S. Kanthak, K. Schütz, and H. Ney, "Using SIMD instructions for fast likelihood calculation in LVCSR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey, jun 2000, pp. 1531–1534.