CONTEXT-INDEPENDENT PHONEME RECOGNITION USING A K-NEAREST NEIGHBOUR CLASSIFICATION APPROACH

Ladan Golipour, Douglas O'Shaughnessy

INRS-EMT, Quebec University, Montreal, Canada

golipour@emt.inrs.ca, dougo@emt.inrs.ca

ABSTRACT

In this paper we investigate a non-parametric classification of English phonemes in speaker-independent continuous speech. We employ the "voting" k-Nearest Neighbour (k-NN) classifier, a powerful technique in pattern recognition problems, along with a new representation of phonemes for the speech recognition task. We also exploit the idea behind "approximate" k-NN that results in a very fast way of computing the k approximate closest neighbours of each data point. Comparing the recognition performance of the proposed method with the HMM-based recognizer of HTK toolkit reveals that the k-NN-based recognizer outperforms its counterpart. In addition, incorporating the "approximate" nearest neighbour search instead of the "exact" one results in completing the training step much faster than the HMM-based system, and the testing step with a comparable computational time. We also reduced the amount of the training data by applying a pattern recognition technique, called "thinning" algorithm. The outcome was a considerable reduction in the k-NN search space and hence the execution time, and also a slight increase in the recognition performance.

Index Terms— speech recognition, pattern classification, approximate index search

1. INTRODUCTION

Conventional speech recognition systems are usually parametric. This means that speech recognition is performed by evaluating input speech samples against an abstraction like a function or a density distribution that models the training data. In parametric speech recognition processing, computational burdens are reduced and speech recognition calculations become more manageable. However, there is a wide degree of variability in the way a given word can be pronounced. Mostly, two acoustic realizations of the same word, or even the same sound, are not identical. Moreover, speech from different speakers can be very different, since speakers differ in their vocal tract length or shape of the articulators, as well as in their speech style (e.g., speaking rate) and regional dialect [9]. In speaker-independent speech recognition systems, the speech models have to accommodate all this variability. Therefore, fuzzy, or poorly defined, word models can be produced which in turn results in recognition errors during the decoding procedure. This fuzziness can also exist in speaker-dependent speech models but in a lesser degree due to the inherent variability of the speech models and of the speech itself. Using the statistical models represents a potential loss of the information in the recognition process. Those details that are present in individual data samples are sacrificed in order to pool information in a controlled fashion [7]. On the contrary, in a non-parametric speech recognition process, different training observations are not blended into one model. Indeed fine phonetic details of the actual utterances rather than statistical approximations can be used during comparison of the input sample against training observations, and accordingly in the recognition process. Incorporating this extra information has the potential to produce more accurate recognition results. As a result, in pattern classification problems, a debate is going on between storing the individual data samples (utterances for speech recognition) and deriving an abstract representation from the training material. A similar debate also exists in the psychological literature related to the human memory [13] (and to Human Speech Recognition (HSR) [3]). Researchers are attracted to the concept of "episodic memory", and there is some evidence that individual memory "traces" are stored. A few researchers have already realized the potential relevance of this topic to Automatic Speech Recognition (ASR), and some work is moving in the direction of at least incorporating non-parametric approaches (e.g., Template-based) in the parametric ones (HMM-based) for large-vocabulary continuous speech recognition (LVCSR) [11].

2. K-NEAREST NEIGHBOUR CLASSIFICATION

In the parametric density estimation approach, a parametric probability distribution is chosen to fit the data, and the parameters of this distribution are estimated in such a way to provide the best fit according to some defined criteria. Generally, the choice of the distribution is based on the prior knowledge about the data (very often based on the central limit theorem; this explains the popularity of the normal distribution), on goodness-of-fit criteria, and sometimes on the belief that the distribution is flexible enough and is able to model any real data distribution accurately. The Gaussian Mixture Model (GMM), which is the most popular choice in HMM-based speech recognition, is of the last category. However, in non-parametric approaches the density estimate is calculated based only on the genuine characteristics of the data, generally according to the information that is derived from the close neighbourhood of the query point itself. In fact, the common practice in HMM-based speech recognizers, which is using a large number of mixtures per distribution, makes the nonparametric approaches more reasonable than the parametric ones for the speech recognition purposes. We intend to apply a non-parametric classification approach and perform context-independent phoneme recognition. One of the most well known such classifiers is the k-nearest neighbour (k-NN). In the "voting" k-nearest neighbour classification, we classify an unknown sample to the majority vote between the class labels of its k closest neighbours in the training dataset. This is a very powerful rule due to the fact that, for large enough training datasets, the error rate is upper bounded by twice the Bayes error rate (optimal error rate). It has also been shown that the gap between its error rate and the optimal Bayes error is directly linked to the value of k, assuming that a large enough training dataset is available [1], [2]. This is an important advantage compared to the continuous estimates such as the Gaussian mixtures. For the continuous estimates, we rely on the assumption that their basic functions are consistent with the true data distribution, but the acoustic vectors generally do not follow Gaussian distributions [8]. In addition, although obtaining a Gaussian mixture density with densities that are

arbitrarily close to the true distribution is theoretically "possible", the theory does not illustrate how we should produce this mixture. For instance, the number of mixtures and the training of the weights in the GMM is actually guided by heuristics, and there is no proof showing the convergence of the distribution model towards the true distribution. As a result, the k-NN probability estimate [1] might be able to offer a convincing alternative to the continuous estimates. The k-NN probability estimate is used in the "volumetric" k-NN classification formula [1]. In this paper we use a similar version of this classifier, called "voting" k-NN, which is already described above.

3. FEATURE EXTRACTION

For accomplishing any recognition procedure, the first step is feature extraction, which provides a reasonable representation of the speech data. The features we extract from speech utterances are the well known Mel Frequency Cepstral Coefficients (MFCCs). We apply the same method in the feature extraction section of the HTK toolkit and compute 14 dimensional MFCC features for frames of 20 msec of speech. In conventional HMM-based speech recognizers, the recognition procedure is repeated for each frame of speech separately, therefore each frame has a representation of usually 42 dimensions that includes MFCCs, delta, and delta delta coefficients. In our approach, we assume that phoneme boundaries are available for the train and the test datasets (these boundaries are already given in the TIMIT database), and divide the duration of each phoneme into three sections. We average the MFCC coefficients of the frames in each part, and generate a 42 dimensional feature vector for each phoneme by concatenating the MFCC coefficients of these three sections. In this way we are able to include time-dependent information of each phoneme in its representing vector as well. As was described before, in the typical non-parametric approach to classification in instance-based learning, training data is collected and used to design a classifier. In the k-NN classification approach, the training data acts as a feature space in which we evaluate the unknown test samples and seek their nearest neighbours. Since we are planning to use Euclidean distance in this feature space, we need to standardize the training data. Otherwise, if one input has a large mean and variance and another a small one, the latter will have little or no influence on the result. Since standardizing the data usually leads to losing some information, sometimes little help might be achieved from this act. In our recognition procedure, we noticed that subtracting the mean for MFCCs enhances the recognition performance while dividing by the standard deviation worsens the performance; therefore, we limit the standardization of the data to subtracting the mean. One other option would be fully standardizing the data and then incorporating some weighting scheme for the features.

4. PHONEME RECOGNITION

In the past the nearest neighbour classification has been unfairly criticized by some mistaken assumptions. For instance, it has been believed that one must store all the data for implementing this rule, or moreover, for classifying an unknown sample, one has to compute the distances between the unknown vector and all members of the training dataset. In fact, these assumptions are not correct and computational geometric progress and faster and cheaper hardware revealed that the k-nearest neighbour rule can be easily practiced in for pattern recognition applications [10]. Due to the fact that for many practical pattern recognition problems, the data rarely has a specific underlying distribution, the exact techniques developed for nearest neighbour search do not improve over sequential search substantially. Therefore, recent research is looking for powerful "approximate" techniques for nearest neighbour search. In order to compute k nearest neighbours of each point in the feature space, we use the algorithm, "approximate multi dimensional index structures" that was proposed by Houle [5]. The previous work in nearest neighbor search algorithms was mostly based on the more traditional tree-based index structure schemes. The main idea of these tree-based schemes was assigning items to the subtree of their nearest node from a limited set of candidates. However, occasionally this process leads to having some nodes whose nearest neighbors can be reached only via very long paths. For more details see Houle [5]. The data structure proposed by Houle counters this problem by allowing multiple paths between the nodes. Unlike a tree-based structure, this structure lets nodes have multiple parents. This results in having very few nodes whose nearest neighbours are reachable only through long paths. Therefore, one obtains a more compact search space by applying this algorithm. During the search, approximate nearest neighbors are located recursively within a large sample of the data and the links from these sample neighbours are followed to discover approximate neighbours within the remainder of the set.

Using the feature extraction algorithm described in the the previous section, the features representing each phoneme in the training and test datasets are computed in 42 dimensions. Then, employing the search algorithm proposed by Houle we build the SASH (Spatial Approximation Sample Hierarchy) [5] from the feature vectors of the training dataset. This step can be equivalent to the training step in an HMM-based recognition system. However, compared to the training part of the HMM-based approach, the construction of the SASH is very fast. This can be seen in the timing information provided in Table 1. For the testing section, we search for the k "approximate" nearest neighbours of each test sample, applying the search algorithm designed by Houle on the SASH, which is already built in the training section. We choose the class (phoneme type) with the highest number of members among the k nearest neighbours of the test point as the recognized value for the test point. The data we used for the recognition process is composed of the training and the test parts of the TIMIT database. The number of phonemes in the training dataset was around 150000 and the amount of the test phonemes was around 54000. We also decrease the number of phoneme classes in use in the TIMIT database to 41 phonemes, by merging the silences of the stops into the stops and deleting some phonemes that are not commonly used in the recognition tasks. In order to compare the proposed k-NN recognition approach with the conventional HMM-based approach, we use the HTK toolkit and perform an HMM-based recognition with MFCC features on the TIMIT database. For this purpose the number of mixtures is put to 5 and 42 dimensional feature vectors are used, including the delta and accelerating coefficients. We also supplied the recognizer with the phoneme boundaries both during training and test processes to maintain equivalent information with the k-NN approach. We do not include any context-dependent information such as biphone, or triphone models, or any language model in both the

Table 1. *The percentage of correctly recognized phonemes of the TIMIT test database for the HMM-based, and the k-NN recognition, approaches.*

	HMM-based	k-NN-based
Correct(%)	56.03	61.1
Training execution time (s)	1925	102
Testing execution time(s)	165	400
Total execution time(s)	2090	502

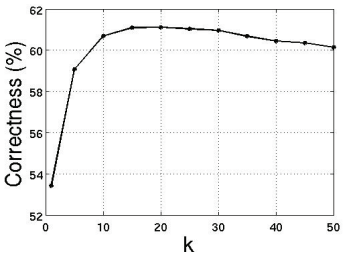


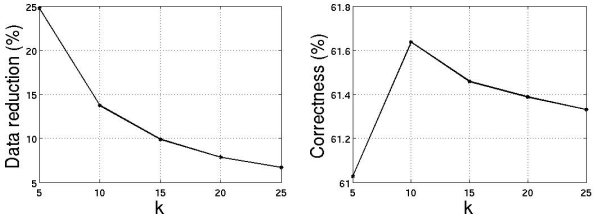
Fig. 1. The correctness (%) vs. the value of "k" in the k-NN-based approach.

HMM-based, and k-NN-based, recognition procedures. The performance is evaluated by the percentage of correctness. This value is computed as the number of correctly classified phonemes in the test dataset divided by the total number of phonemes. Table 1 shows the performance of both the k-NN-based and the HMM-based recognizers. As can be seen, the k-NN-based recognizer outperforms the HMM-based system. As is known, HTK uses Gaussian mixtures for modeling the phonemes, and it has been claimed that using only the distance between the test sample and the training data is too simplifying, but the result clearly contradicts this belief. Table 1 also shows the average execution time over each run of the training and test steps for both recognition systems in C++ v4.1.1 and under Linux Fedora with a single 3.2 GHz Pentium IV. The results show that surprisingly, using the "approximate" version of the nearest neighbor search, and also processing phonemes rather than 20 ms frames of speech signal (which is applied in HMM-based recognizers), both the training and the testing steps for the k-NN-based recognizer are completed much faster compared to the HTK toolkit using the same processing machine. There are some results on phoneme classification using the TIMIT database, in which phoneme boundaries have been used. For example, Young used 48 context-independent models, and achieved 61.7% correctness, for a test sequence of 160 sentences randomly taken from the training sequence, [6]. Chengalvarayan achieved 65.66% correctness for a proposed time-varying discriminative training technique effective for phonetic discrimination for 39 context-independent models and 5 Gaussian mixtures, [6]. Wachter [11] also examined the k-NN classifier on the TIMIT database, but he performed "frame" classification, and he reported 63.5% using the Euclidean distance as the distance measure and no pre-processing of the data (similar conditions to our approach). Our method has comparable results with these Benchmark HMM methods, and in addition it is very fast. For the k-NN classification task, one should choose a proper value for k according to the nature of data being in use. For this purpose, different values of k ranging from 1 to 50 are examined, and the result is depicted in Figure 1 . As it is seen, the values of k between 15 to 20 are the optimum choices for the clean speech signal of the TIMIT database.

5. THINNING THE TRAINING DATASET

Since in the k-NN classification the training data should be used for the classification task, even if we incorporate sophisticated search algorithms, there is still a vast search space in relatively high dimensions. As a result, many researchers proposed methods that reduce the amount of the training data without sacrificing the performance, and therefore storing and processing as little of the training data as possible. These methods are called "thinning" or "condensing" methods in the pattern recognition literature, [4]. The condensing method that we use is the k-NN version of the thinning procedure:

1. Compute the k nearest neighbours of each point in the training dataset.
2. Mark those points that all their k nearest neighbours are not from the same class as the point itself.
3. Delete the marked points.



(a) (b)

Fig. 2. (a) The amount of data reduction (%) vs. the value of "k" when applying the thinning algorithm. (b) The percentage of correctness vs. the value of k after applying the thinning algorithm.

The idea behind this algorithm is very intuitive and simple. Basically, those points in the training dataset that are mostly surrounded by the members of their own class, and are far from the boundaries between classes, are removed. This method of decreasing the number of points in the training dataset does not reduce the performance of the system since the omitted points are usually located far from the boundaries between classes, and therefore do not have crucial roles in the k-NN classification decisions. We apply this algorithm on the original training dataset before using it for the classification of the unknown test phonemes. Clearly, the amount of reduction of the training dataset depends on the chosen value for k. For instance, if k is large, we will have a small number of deletions since there will be a very limited number of points for which a large number of neighbors are from the same class as the point itself. As the value of k decreases, more points are omitted in the process of thinning and this might affect the classification performance. We apply the thinning algorithm with different values of k and achieve the results displayed in Figure 2. The best value for k is 10, which reduces the training dataset by 13.7%. Surprisingly, the performance of our recognition algorithm increases slightly to 61.6% when we apply the thinning algorithm with $k = 10$ on the training dataset. This can be due to the fact that this process helps to reduce the effect of the bias induced by more populated classes while it preserves the decision boundaries between classes.

6. CONFUSION MATRIX

For demonstrating the performance of a speech recognizer, usually measures like the "percentage of correctness" or the "accuracy" are taken into account. Although these measures supply the user with a reasonable impression about the performance of the recognizer, they are quite general and provide little information about how the recognizer acts for different phonemes with various intrinsic characteristics, and consequently give little insight about the potential ways of improving the recognizer's performance. In order to compare the behaviour of the k-NN-based and the HMM-based recognizers more deeply, we decided to compare their confusion matrices. The differences between the two confusion matrices are quite interesting. In the HMM-based system, the recognition errors for a specific phoneme usually spread over all phonemes with different degrees, while for the k-NN case we have spike-like errors for

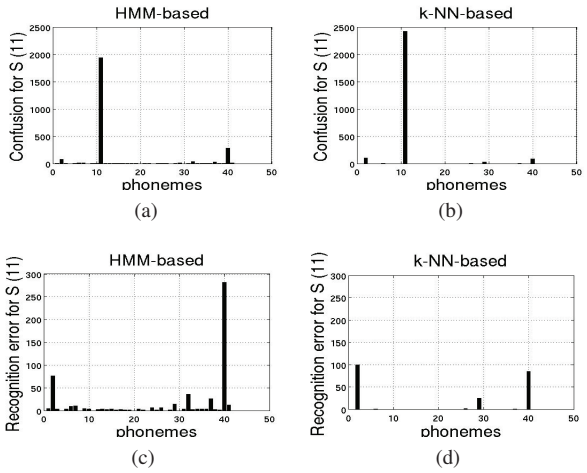


Fig. 3. (a) The confusion in the recognition of phoneme "s" using the HMM-based approach. (b) The confusion in the recognition of phoneme "s" using the k-NN-based approach. (c) The recognition errors of phoneme "s" using the HMM-based approach. (d) The recognition errors of phoneme "s" using the k-NN-based approach.

some phonemes and usually zero confusion for the others. Figure 3 shows the recognition errors for phoneme "s" in the HMM-based and the k-NN-based approaches. The amount of correctly recognized "s"'s are also displayed for both systems in this figure. We can interpret the above occurrence as follows. For the k-NN classifier, those classes that are close neighbours of a particular class (here, "s") and share main decision boundaries with it, produce the spike-like confusions, and the rest of the classes are hardly confused with the expected one. For the HMM-based system, the acoustic model that represents a particular phoneme can produce the same log likelihood for other phonemes due to the general parametric model in use (GMM). We observe one shortage of the k-NN-based approach. Although the population of each phoneme in the training dataset represents the prior probability of that phoneme and considering it during the recognition task improves the performance, inspecting the confusion matrix for this approach reveals that the phoneme classes that have a very high population (e.g., "n" with around 8100 members) in comparison with their neighbouring classes (i.e., those with relatively low populations) (e.g., "ng" with around 1300 members) bias the recognition result towards themselves. Since this situation occurs only for classes with very low prior probabilities, the recognition performance still remains better than the HMM-based system. In order to lessen such an effect, one possibility can be applying the weighted k-NN classification, with weights inversely proportional to the population of each class, or related to the distance of the query point to its neighbours, having higher weights for the close neighbours than those further away. We plan to apply these methods in the future to overcome the described shortage of the k-NN-based system.

7. CONCLUSION

Although template-based speech recognition is still in its early stages, in comparison to the widely used conventional algorithms like HMM-GMM-based speech recognition, the results achieved in this direction by various researchers are quite promising. This can be due to its inherent similarities with the human speech recognition, and also to the high variability of the speech data that can hardly be modeled accurately enough using the parametric estimates. Therefore, we were encouraged to step in this direction, and propose a new method that employs a powerful and intuitive non-parametric classifier, voting k-NN, a very fast approximate search approach, a new representation of phonemes, and finally a thinning algorithm for reducing the training dataset. The recognition result shows a promising increase in the percentage of correctness over the conventional HMM-based phoneme recognition. In addition, applying the approximate nearest neighbour approach for the classification purpose rather than the exact one leads to achieving a very lower training execution time compared to the HMM-based system, and also a comparable execution time for the testing. In the future, we plan to overcome the shortages of this approach, and devise a method of integrating it in conventional methods of speech recognition.

8. REFERENCES

[1] Fukunaga K., and Hostetler L., "k-nearest-neighbor bayes-risk estimation," *IEEE Transactions on Information Theory,* 1975, vol. 21, no. 3, pp. 285-293.

[2] Fukunaga K., and Kessell D., "Nonparametric Bayes error estimation using unclassified samples," *IEEE Transactions on Information Theory,* 1973, vol. 19, pp. 434-439.

[3] Goldinger S. D., "Words and Voices: Episodic Traces in Spoken Word Identification and Recognition Memory," *Journal of Experimental Psychology: Learning Memory and Cognition,* 1996, vol. 22(5), pp. 1166-1183.

[4] Hart, P. E., "The condensed nearest-neighbor rule," *IEEE Transactions on Information Theory,* 1968, vol. IT-4, pp. 515-516.

[5] Houle M. E., and Sakuma J., "Fast approximate similarity search in extremely high-dimensional data sets," *Proceedigs of the 21st International conference on Data Engineering(ICDE2005),* 2005, pp. 619-630.

[6] Klautau A., http://www.laps.ufpa.br/ aldebaro/papers/Timitresults.pdf.

[7] Maier V., and Moore R. K., "An Investigation into a Simulation of Episodic Memory for Automatic Speech Recognition," *Proc. Eurospeech 2005,* 2005.

[8] Montacie, C., Caraty, M.-J., Lefevre, F., "K-nn versus Gaussian in a HMM-based system," *Proceedings ESCA Eurospeech,* 1997, vol. II, pp. 529533.

[9] Sharenborg O., Norris D., Bosch L., and McQueen J. M., "How Should a Speech Recognizer Work?", *Cognitive Science,* 2005, vol. 29, pp. 867-918.

[10] Toussaint G. T., "Proximity graphs for nearest neighbor decision rules: recent progress," *Interface-2002-4th Symposium on Computing and Statistics (theme: Geoscience and Remote Sensing),* 2002.

[11] De Wachter M., "Example Based Continuous Speech Recognition", PhD thesis, K.U.Leuven, ESAT, May 2007.

[12] De Wachter M., Matton M., Demuynck, K., Wambacq P., Cools R., and van Compernolle D., "Template-based Continuous Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* 2007, Vol. 15(4), pp. 1377-1390.

[13] Yotsumoto Y., Wilson H. R., Kahana M. J., and Sekuler R., "Episodic Recognition Memory for High-Dimensional, Human Synthetic Faces," *Journal of Vision,* 2003, vol. 3(9), pp. 683, 683a.