DECISION RULES FOR SPEAKER-INDEPENDENT ISOLATED WORD RECOGNITION

Vishwa N. Gupta, Matthew Lennig, and Paul Mermelstein

Bell-Northern Research
3 Place du Commerce
Nuns' Island, Verdun H3E 1H6 CANADA

ABSTRACT

This study compares the recognition rates attainable with the aid of two different methods of generating reference templates from training words and two different decision rules. The test enviroment consists of isolated words from a small vocabulary spoken by a large number of speakers over the public telephone system. Experiments performed show that the use of individual word templates for references together with the k-nearest neighbor decision procedure substantially improves the performance in isolated word recognition. We attempted to minimize the computations involved in the k-nearest neighbor decision procedure by assuming that the dynamic time-warp distance was a metric, which would allow use of a 1-nearest neighbor decision rule with appropriately relabeled reference data. Results indicate that this step leads to an error rate exceeding that obtainable with the 1-nearest neighbor rule on the original nonrelabeled data.

INTRODUCTION

In order to attain speaker independence in word recognition, training data from a large number of speakers are required. Computational costs resulting from such large training sets are typically reduced by clustering training data of each word into a given number of clusters [1,2]. An average template is then derived from each cluster.

Two arguments can be given against the use of average templates and in favor of using all the training tokens as individual templates. First, the process of averaging requires time alignment of the frames of two tokens before they can be averaged. The aligning of the frames of the tokens is done automatically by a dynamic time warping algorithm which may result in frame misalignments and smearing of acoustic information when feature parameters of frames corresponding to two different sounds are averaged. This smearing of information contained in the tokens may lead to a lower recognition performance. The second argument against the use of the average templates is that the decision boundaries estimated on the basis of a set of average templates are only approximations to those that may be estimated from the aggregate of training tokens. Thus, one question to be asked is whether a loss of recognition accuracy occurs by the use of average templates.

RECOGNITION EXPERIMENTS

Experimental Procedure

The vocabulary used in the experiments consisted of the ten digits and four control words: "yes", "no", "stop", and "repeat". The training set consisted of approximately 130 tokens per word collected from a total of 55 male and 55 female speakers. The test set consisted of a total of 749 tokens collected from 15 male and 15 females speakers. Training and test speakers were distinct. All were native speakers of English. The words were captured over public dialled-up telephone lines. Speech data are represented in terms of seven mel-based cepstrum parameters computed at intervals of approximately 19 ms. The preprocessing and dynamic programming algorithms used in these experiments are those described in [1]. Clustering was accomplished using a complete-link clustering algorithm.

Experiments with the Nearest-Neighbor Rule

Two experiments were carried out using average reference templates. In Experiment 1, training tokens were clustered into six clusters per word, yielding six average reference templates per word. The nearest neighbor decision rule in conjunction with these templates resulted in 46 errors (6.1 percent error rate).

Experiment 2 was identical to Experiment 1, except that 12 reference templates per word were used. This resulted in 35 errors (4.7 percent error rate).

The results of Experiments 1 and 2 seem to indicate that recognition can be improved, given a fixed training set, by increasing the number of templates per word. In Experiment 3, we carried this hypothesis to the extreme by using each

training token as a template. As in Experiments 1 and 2, the nearest-neighb decision rule was used. The result, 35 errors (4. percent), was identical to that of Experiment 2. hus, a limited amount of averaging can reduce computation time without impairing performance.

Experiments with the k-Nearest Neighbor Rule

The k-nearest neighbor rule provides a more reliable estimate of the likelihood that an unknown belongs to a particular category by taking into account more than one reference token in the neighborhood of the unknown.

In Experiment 4, the reference templates comprised all the training tokens without any averaging. The k-nearest neighbor decision rule was used to recognize the test tokens, using the 1-nearest neighbor rule to break ties. For each value of k, the 1-nearest neighbor decision was necessary for only a small fraction of the test tokens (seven tokens or less out of 749). We computed recognition results for values of k ranging from 1 to 13. Table 1 shows the recognition results for this experiment. The recognition rate peaked for values of k between 7 and 13, and the total number of errors reduced to 24 (3.2 percent error rate).

Conclusions from the Recognition Experiments

Use of the k-nearest neighbor decision rule with individual training tokens reduces the number of errors by roughly one third as compared with the results of Experiments 2 and 3. We conclude that one or both of the arguments against averaging discussed in the introduction holds. Because the improvement in recognition rate is substantial, it would be desirable to use the k-nearest neighbor technique if the computational drawback could be removed.

TECHNIQUES FOR REDUCING THE K-NEAREST NEIGHBOR COMPUTATIONS

The k-nearest neighbor algorithm is computationally expensive since it requires one distance calculation from each training token to the test token. A number of methods may be used to reduce the required computations. For some of these methods to be applicable, we need to assume that the dynamic time warp distance is a metric. The dynamic time warp distance is positive and symmetric, however, it does not always satisfy the triangular inequality. We ran some experiments to see how often the triangular inequality is satisfied. Out of the 3.3 million triplets tested, we observed only 150,000 exceptions (less than 5%). This suggests that we can use the following algorithms to reduce the number of computations involved in the k-nearest neighbor computations.

If we assume that the dynamic time warping distance is a metric, we can implement the k-nearest neighbor decision rule with a 1-nearest-neighbor algorithm by a relabeling of the training tokens [4]. For each training token, we compute its k-nearest neighbors. We relabel the token with the label of the word in majority among the k-nearest neighbors. In the test phase, the 1-nearest-neighbor algorithm can be then used to implement the k-nearest neighbor decision. Once the k-nearest decision procedure has been reduced to the 1-nearest neighbor algorithm, we can apply the triangle inequality to reduce the number of comparisons necessary to find the nearest neighbor [5].

Another method to achieve computational savings is to implement the nearest neighbor decision as a binary search [6,7]. To achieve this, training tokens are associated with the leaves of a binary tree. Starting at the root, a binary decision is made at each node until a leaf is reached. Each binary decision is based on distance computations to two average or representative tokens, representing the aggregate leaves of their corresponding subtrees. In order to guarantee finding the nearest neighbor, certain training tokens must be multiply represented in the leaves of the tree, resulting in somewhat greater than O(log N) distance computations.

RELABELING EXPERIMENT

The result mentioned in the previous section implies that the dynamic time warp distance is almost a metric: It obeys the triangle inequality 95 percent of the time. To understand how this result impacts recognition performance when the nearest neighbor algorithm is used to implement the k-nearest neighbor decision rule, we ran the following experiment.

In a preprocessing stage, each training token was relabelled according to the k-nearest neighbor rule (k was varied experimentally from 7 to 19). About 4 percent of the training tokens were relabelled. In the recognition stage, the nearest neighbor rule was used with the relabelled training tokens. Table 2 shows the results as a function of k.

CONCLUSIONS

We can significantly improve the performance of the isolated word recognizer if we use the individual word tokens as reference templates with the k-nearest neighbor rule instead of average reference templates. A number of techniques can be used to minimize the computations involved in implementing the k-nearest neighbor decision procedure. However, these techniques depend on the warp distance having metric properties. Although the warp distance was seen to be nearly metric, it

is not sufficiently so to permit effective implementation of the k-nearest neighbor rule with a nearest neighbor algorithm.

REFERENCES

[1] V. Gupta and P. Mermelstein, "Effects of speaker accent on the performance of the speaker-independent, isolated-word recognizer," J. Acoust. Soc. Am., Vol. 71(6), pp. 1581-1587: 1982.

[2] L.R. Rabiner and J.G. Wilpon, "Speaker-independent isolated word recognition for a moderate size (54 word) vocabulary," IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-27, No. 6, pp. 583-587: 1979.

[3] M.J. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," 1980 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 880-883: 1980.

[4] G. Toussaint, B.K. Bhattacharya, and R.S. Toulsen, "The application of Voronoi diagrams to nonparametric decision rules," to appear in Computer Science and Statistics: 16th Symposium on the Interface, Atlanta, March 1984.

[5] C.D. Feustel, and L.G. Shapiro, "The nearest neighbor problem in an abstract metric space," Pattern Recognition Letters, Vol. 1, pp. 125-128: December 1982.

[6] L.G. Shapiro and R.M. Haralick, "Organization of relational models for scene analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-4, No. 6, pp. 595-602: 1982.

[7] L. Miclet and M. Dabouz, "Approximate fast nearest-neighbor recognition," Pattern Recognition Letters, Vol. 1, No. 5 and 6, pp. 277-285: 1983.

Table 1. k-nearest neighbor recognition results for different values of k. Approximately 130 tokens per digit from a total of 55 male and 55 female speakers were used as reference tokens. The test tokens were obtained from 15 male and 15 female speakers.

k =	1	3	5	7	9	11	13
# recognized correctly by k-nn rule	714	714	716	721	722	723	722
# of no decisions	0	7	7	7	5	3	4
1-nn result for no decisions	0	3	4	4	3	2	3
total correct	714	717	720	725	725	725	725
total errors	35	32	29	24	24	24	24
% error	4.7	4.3	3.9	3.2	3.2	3.2	3.2

total tokens = 749

Table 2. Results from relabeling experiment. For each value of k, the number of relabeled training tokens is shown, as well as the recognition error rate obtained using the nearest neighbor rule with the relabeled training data.

k used for relabelling in training phase	7	9	11	13	15	17	19
Number of tokens relabeled in training phase	79	77	74	72	74	70	73
% relabeled	4.4	4.3	4.1	4.0	4.1	3.9	4.1
Number of errors: nearest neighbor, relabeled tokens	40	38	41	38	41	41	40
% error	5.3	5.0	5.5	5.1	5.5	5.5	5.3