Digit Recognition Using K-Means Clustering

Bryan Tran

Abstract

In this study a digit-speech recognition system utilizing the k-means clustering algorithm is presented. Its effectiveness as a function of k is explored and discussed. The TIDIGITS corpus and the Hidden Markov Model Toolkit (HTK) were used to produce two data sets, used for training and testing the system, respectively. Various k values were used in the training phase of the system, and its effects are presented showing a dramatic reduction in error as k increases to 20, followed by very little reduction in error as k continues to increase.

Introduction

In building speech recognition systems, speech signals can be thought of as a "message encoded as a sequence of one or more symbols", where the task of recognition lies in representing these symbols as speech vectors, and then interpreting and mapping these speech vectors into recognizable classes such as words. However, difficulties in neatly mapping these symbol sequences to speech, coupled with the fact that the boundaries between these symbols are nebulous in nature, complicate this effort. Consider also the inherent variability of speech signals due to the speaker's age, gender, mood, sickness etc., and one can see the problems involved in speech recognition are manifold (Young et al., 2006).

In this study, a digit-speech recognition system utilizing the k-means clustering algorithm is presented. The author explored the system's effectiveness, (which was measured in terms of classification error), as a function of the number k chosen in the aforementioned algorithm.

Materials and Methods

Data collection

The TIDIGITS corpus and the Hidden Markov Model Toolkit (HTK) were used to generate the data sets used for training and testing the digit speech recognition system. The TIDIGITS corpus is a large data set of speech files, where each file is a spoken digit (0-9), and 0 is pronounced both as "zero" and "oh". These speech files were recorded from over 300 men and women of various ages and dialects in the continental United States. Meant to evaluate speech recognition algorithms, the corpus is partitioned into two sets: training and testing (Leonard and Doddington, 1993). The HTK, "a toolkit for building Hidden Markov Models", was then used to extract mel-frequency cepstrum files (mfc) from the TIDIGITS corpus, yielding a set of mfcs for training and testing the digit recognition system (Young et al., 2006). Programming was done within the MATLAB computing environment and programming language (Mathworks, Natick, MA).

Training

For each digit class, its training data is extracted from its corresponding mfc data set, and is concatenated into a single training matrix. K-means clustering is then run on this training matrix to produce a codebook of a k number of code words. A brief overview of the k-means clustering algorithm used is as follows (Roch 2015):

1. Randomly select a k-number of training vectors to be the initial centroids.
2. Initialize the overall distortion value ("old distortion") by calculating the mean of the minimum Euclidean distortions between all training vectors and the initial centroids.
3. Partition the training vectors according to what centroid yields minimum distortion with each individual training vector, i.e., which centroid each training vector is "closest" to.
4. Calculate new centroids by taken the average of each partition.
5. Calculate a new overall distortion ("new distortion") value as in step 2.

6. If the ratio of ("new distortion") / ("old distortion") is greater than some user-chosen threshold, exit the algorithm. Else, finish step 7, go to step 3 and continue iterating.
7. Set the old distortion equal to the new distortion.

A single codebook "library" array is then constructed, where each entry of the array is a digit class's codebook. These codebooks will be compared against the testing data and will ultimately be used to make the digit class predictions.

Testing

In the same manner as in the training phase, for each digit class, its testing data is extracted from its corresponding mfc data set. Instead of concatenating all the data into a single matrix however, the system compares each testing vector against the codebooks generated in the training phase. Whichever digit class's codebook produces the smallest mean of the minimum Euclidean distortions against the testing vector is the system's prediction for this testing vector's digit classification. Each prediction is recorded in a 10x10 confusion matrix, where the rows indicate each digit class and the columns indicate the possible classifications for each digit class (where 10 is the zero or "oh" class). For example, cell (5,5) represents all of the correct classifications of the test vectors of digit class 5, while the cell (5,6) represents all of the incorrect classifications of test vectors of digit class 5 into digit class 6.

Results

The digit recognition system was trained and tested eight times with k values of 1, 5, 10, 20, 50, 100, 500, and 1000, respectively. The k values were chosen somewhat arbitrarily, in anticipation of a decrease in error reduction around k = 100. Figure 1 depicts the error fraction of each training and testing cycle against its corresponding k value, where the error fraction is simply the number of incorrect predictions, divided by the total number of predictions.

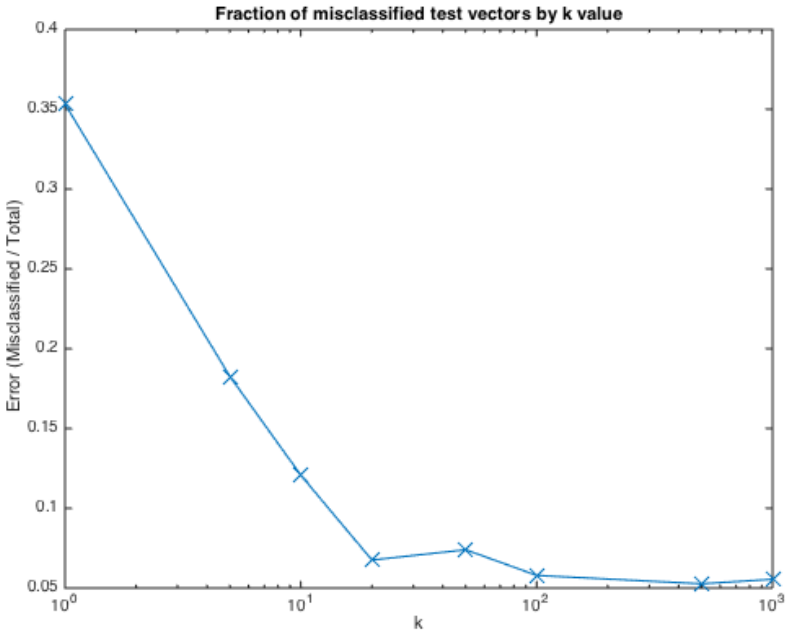

Figure 1. Classification error as k values increase.

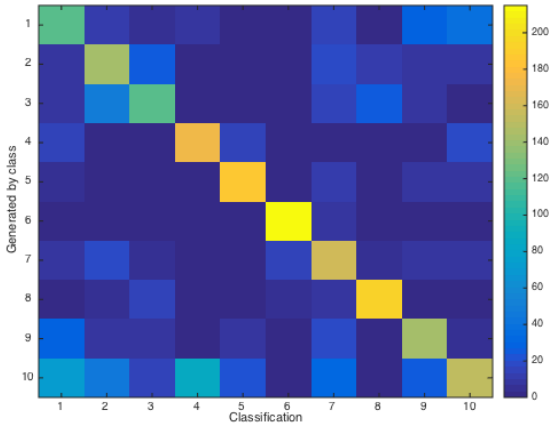

Figure 2. Confusion matrix with k = 1.

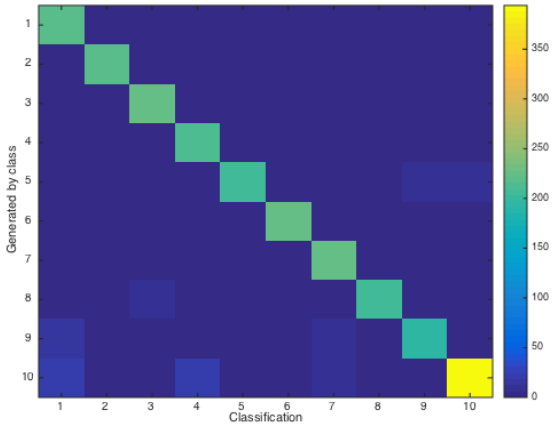

Figure 3. Confusion matrix with k = 20.

Figures 2 – 4 show the confusion matrix visualizations for the training and testing cycles with k means of 1, 20, and 500, respectively. As described before, the rows represent the digit classes and the columns represent the possible digit classifications of each digit class. Therefore, the diagonals of the confusion matrices from top left to bottom right represent the correct classifications.

Discussion

As expected, an increasing k value generally corresponded with a decreasing error fraction. Beginning with a value of k = 1, an increasing k yields a dramatic reduction in the amount of digit misclassification. At k = 1, the system yielded an error fraction of 0.3532, and at k = 20, the error fraction is a mere 0.0676, or a 6.76 percent error. Around k = 20, however, the returns on error reduction begin to diminish with increasing k, and

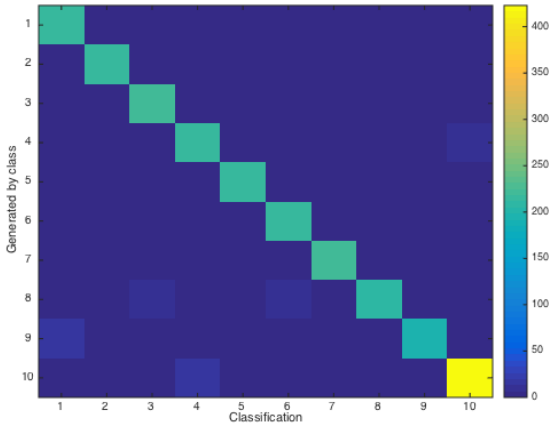

Figure 4. Confusion matrix with k = 500.

the error fractions plateau beyond k = 20. In fact, k values of 50 and 1000 yield slight increases in error (Fig. 1). In visualizing the confusion matrices, one can more vividly see the error reduction between k = 1 and k = 20, and the little effect that increasing k beyond 20 produces (Fig. 2-4). Some areas of confusion persist despite drastic increase of k from 20 to 500. Notably, the cells at (9,1), (4,10), and (10,4) exhibit persisting error. Intuitively, this makes sense, as one and nine share the "ne" phoneme and four, zero, and "oh" share the "o" phoneme. It seems then, that the optimal k value for the system is around 20: only slightly greater than the minimum error at k = 500 (0.0676 versus 0.0527), but also noticeably faster in its performance.

Conclusion

A digit-speech recognition system utilizing k-means clustering was presented and evaluated with the author proposing an optimal k value of 20. With an error of 6.76%, and considering that few steps were taken to optimize the process for speed, it seems reasonable to conclude that a digit recognition system using the aforementioned methods and materials is viable.

Future research should more thoroughly evaluate this system, using both a larger range of and a finer resolution of k-values. In regards to persisting error cases despite increasing k values, e.g. misclassification due to the "ne" phoneme in one and nine, research should be made attempting to resolve this confusion. Lastly, the k-means clustering algorithm used in this study could be replaced entirely with a different vector quantization technique, allowing one to better judge the feasibility of the author's system in relation to its alternatives.

References

Leonard, R. G. and Doddington, G. (1993). TIDIGITS, vol. 2007: Linguistic Data Consortium, Philadelphia.

Roch, M. (n.d.). Clustering Part I. Retrieved October 5, 2015, from http://roch.sdsu.edu/cs682/slides/02ClusteringI.pdf

Young, S., Evermann, G., Hain, T., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. et al. (2006). The HTK book, version 3.4.