Data Preprocessing via Polynomial Fitting to Improve Clustering of Spoken Arabic Digits

Imad Rahal, Preston Hardy and Kelsey Weiers
Department of Computer Science
College of Saint Benedict and Saint John's University
211 Peter Engel Science Center
Collegeville, MN 56321
irahal@csbsju.edu

Abstract

Advancements in speech recognition and voice-to-text technologies have made these topics more popular as of late. When analyzing certain types of time-series data, it is important to properly preprocess the data in order to deal with varied-length instances. In this work we study the influence of three different data preprocessing techniques on the quality of patterns extracted from a dataset of Arabic digits spoken by native speakers. Our ultimate objective is to determine which data preprocessing technique is most effective in producing strong patterns from this dataset which we accomplish by clustering the preprocessed datasets and measuring the quality of the produced clusters. Our experiments underscore the benefits of using the proposed polynomial fitting preprocessing technique to produce desirable results.

Introduction

Time-series analysis is often utilized to expose patterns from data collected over a period of time where the latter plays a significant role in interpreting the semantics of the data. When making comparisons regarding data of any form, it is vital that the various instances in the dataset be scaled into the same proportion as the others. Particularly important to our work, time-series data of different lengths need to be converted to the same spectrum in order to be more readily comparable. In considering time-series analysis of digitized spoken words, it is recognized that data collected over a varied length of time with differing frequencies will have a non-uniform structure. As such, preprocessing the data is usually sought to aid data mining and machine learning algorithms to be more effective in extracting desirable patterns.

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper we propose three techniques to preprocess a dataset of spoken Arabic digits in order to deal with the varying lengths of the instances in the dataset. The most promising technique works by fitting all the values in each instance attribute with a polynomial over time, and then stretching the resulting polynomial to fit the largest sized data piece in a process known as polynomial fitting. Another preprocessing technique is to keep track of the number of increasing and decreasing segments among the values of each instance attribute. The resulting count is then used as a representative value for that particular instance attribute; a similar process is then repeated for all attributes in the dataset. The final preprocessing technique is simply to compute and use the average of the different values per attribute.

Clustering is then applied on the resulting preprocessed datasets. The purities of the produced clusters are studied by analyzing the entropy (Quinlan 1986; Tan, Steinbach, and Kumar 2005) of the clusters in an attempt to understand the tendency of the data instances to cluster in adherence to the spoken digits they represent.

The same dataset was used in two previous studies for speech recognition purposes. This is in contrast to our work which aims to study the effect of different preprocessing techniques on the quality of patterns extracted from the dataset. Hammani and Sellam (2009) used a spanning-tree-based classifier for speech recognition. The data was fed to a model which used class probability to predict the spoken digit. A more efficient version of this approach is presented in Hammani and Bedda (2010).

Numerous works have focused on the problem of speech recognition for various purposes such as improving services for the disabled (Garrett et al. 2011; Zahorian, Zimmer and Meng 2002), robot control (Shrivastava et al. 2013), speaker identification (Beigie 2011) and many

more. Research has been applied to different spoken languages including Urdu (Sarfraz et al. 2010) and Russian (Vazhenina et al. 2012).

Data and Preprocessing

The dataset investigated in this work is called "Spoken Arabic Digit Data Set" and can be obtained from the UCI Machine Learning Repository (Bache and Lichman 2013). The dataset contains 8,800 records. Each record corresponds to a native speaker speaking one of the Arabic digits 0 through 9; in total, there 88 speakers each speaking the 10 digits 10 times for a total of 8,800 records.

The data is split into the following: 66 speakers belong to the training set and 22 speakers belong to the test set; thus, the training set contains 6,600 records and the test set contains 2,200 records. There are no missing attribute values in this dataset. It is worth noting that of the 88 speakers, half are females and half are males.

Each time line (or simply line) in this dataset corresponds to a time-series instance called an analysis frame which represents a time snippet and is made up of exactly 13 attributes corresponding to Mel Frequency Cepstral Coefficients (MFCC) (Zheng, Zhang and Song 2001) in increasing order. These lines are organized into blocks called records ranging anywhere from 4 to 93 lines (or time snippets) of 13 attributes each where a block or record of lines represents the pronunciation of a single Arabic digit by a single speaker. The variations in the number of time snippets per record are largely due to the fact that people tend to speak with different pace and pitch.

The first step in preprocessing this dataset was to remove records at the extreme ends with either very few time lines or with the most lines with the objective of eliminating outliers that might otherwise skew the data. We chose to only include records having between 20 and 60 lines which eventually left most of the dataset intact. The dataset was then preprocessed using the three different techniques explained next in order to convert records of different lengths to the same spectrum.

Averaging

This preprocessing technique entails computing the average value of each attribute across all the record lines. Thus each record is consolidated down to precisely 13 attributes followed by a final entry corresponding to the class label represented by the spoken digit.

Counting Increasing/Decreasing Segments

Using this technique, we compute the number of increasing and decreasing segments per record attribute. This is done by keeping track of the direction of increase/decrease in the attribute value across the record lines and then adding one to a count anytime there is a switch. Figure 1 shows 17 values for some record attribute connected by a curve and how this would correspond to an increase/decrease count of 5. Thus, like the previous technique, this one also condenses each attribute into just one, therefore, producing records condensed into 13 attributes plus a class label.

Polynomial Fitting

This technique fits the data values in every record attribute–ranging between 20 and 60 values–with the unique polynomial that passes through the given values. For example, assuming a given record contains 25 time snippets, then for each of the 13 attributes, we would use the 25 data values $(0, v_0), (1, v_1), \ldots, (24, v_{24})$ to find the unique 24th degree polynomial (call it $f(x)$ where x represents the time snippet and is defined over [0, 24]) that passes through these 25 data values. In other words, we need to solve a system of equations to find the coefficients for polynomial $f(x) = a.x^{24} + b.x^{23} + c.x^{22} + \cdots + w.x^2 + y.x^1 + z.x^0$ given that $f(0) = v_0, f(1) = v_1 \ldots f(24) = v_{24}$.

After deriving $f(x)$, we stretch f so that x is defined over [0, 59] instead of [0, 24] since the number of values per attribute (or lines per record) maxes out at 60. Thus we want to stretch f horizontally by a factor of 59/24≈2.458. This can be achieved by computing function g as $g(x) = f(x/(59\ /\ 24)) = f((24/59)x)$ where $g(0) = f(0)$ and $g(59) = f(24)$. This step is done so that all record attributes are defined for any x in the range [0, 59] (i.e. have up to 60 values).

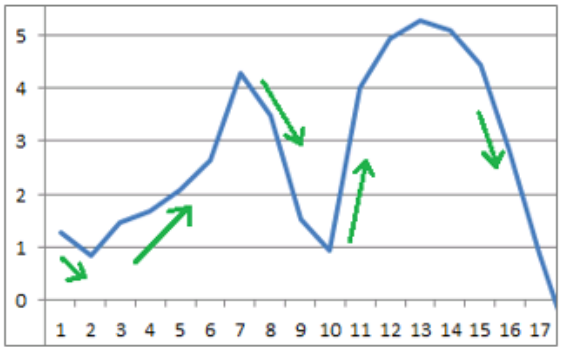

Figure. 1: *Distribution of 17 attribute values with arrows indicating increase/decrease in value. The resulting count is 5.*

The top portion of Figure 2 shows two polynomials defined over different ranges. Notice how the darker polynomial has been stretched to better align with the other one in the lower portion of the figure.

Once attribute polynomials are obtained, there are several ways they can be represented in the data. We chose to evaluate the polynomial at evenly spaced intervals of 5 and treat these values as derived attributes. Thus, for a given record, there would be 13*12=156 attributes in addition to

the class label. The value 13 comes from the original 13 attributes and the 12 is the size of the range (i.e., 60) divided by the size of the interval chosen to evaluate the stretched polynomial (i.e., 5). Thus, one record in the preprocessed dataset would look like the following: $f_0(0),\ f_0(5), \ldots f_0(55), f_1(0), \ldots f_{12}(55)$, *class label*.

Note that an interval of 5 was chosen here in order to balance overall runtime performance with the quality of the produced results. A smaller interval would result in more attributes thus producing better results (most likely) but would also require longer runtimes.

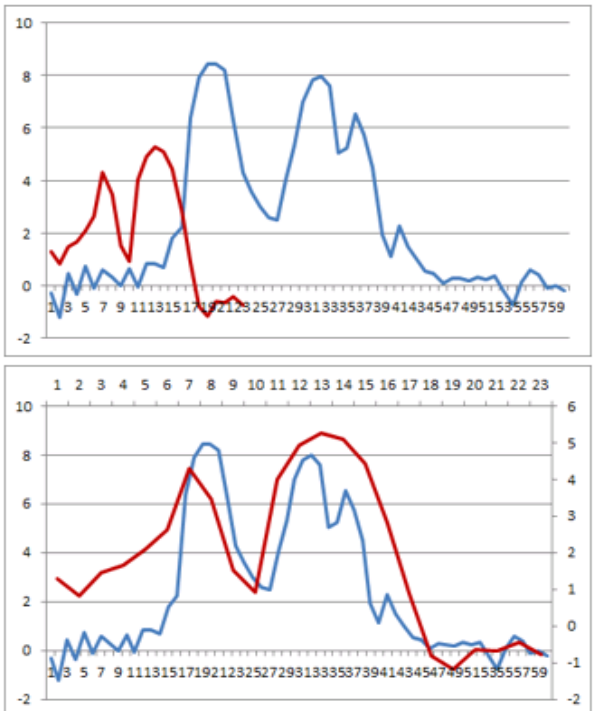

Figure 2: *Adjusting polynomials of differing sizes via stretching.*

Experimental Results

The previous section outlines the preprocessing work done on the dataset; however, our objective in this paper is to experimentally discover which preprocessing technique produces the "best" clustering results. Consequently, we produced clusters for each preprocessing technique and evaluated the goodness of the produced clusters using information gain.

Due to its popularity, we opted to use the traditional K-Means clustering algorithm to cluster the preprocessed datasets. K-Means is a simple partitional clustering algorithm where each cluster is defined by a center point known as the centroid for that cluster. There is a predetermined number of centroids and thereby a predetermined number of resulting clusters (the user-specified value K). Every data point is assigned to a cluster based on which centroid is closest to it using some predefined distance/similarity measure. The algorithm begins by randomly selecting K points as the initial centroids. K clusters are then formed by assigning every point to its closest centroid. The centroids are recomputed, and the points are reassigned until the centroids do not change (or do not change enough). After the clusters stabilize, the algorithm is said to have converged.

We evaluated the produced clusters by computing the entropy defined by information gain (Quinlan 1986). Information gain (or simply info gain) is a measure typically used in decision tree induction (DTI) (Quinlan 1986) during the process of building the decision tree The following formula shows the gain in information from a given split as the reduction in entropy due to that split. This is computed as the current entropy (prior to the split) minus the weighted summation of the entropy values for all tree branches i (where $i \in [1, k]$) resulting from the split where n is the total number of instances before the split and n_i is the number of instances along branch i: $GAINsplit = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i)\right)$.

For every branch i resulting from the split, entropy is computed using the following formula where $p(c|i)$ is the probability for an instance with class label c to exist along branch i:$Entropy(i) = -\sum_c\ p(c|i) \log p(c|i)$.

In the context of data clustering where the data already contains class labels (the spoken digits in our case) one can compute the information gain that results from clustering the data by viewing every cluster as a new branch in some decision tree. Note that higher info gain values indicate higher purities and thus better clustering.

Results depicted in Figures 3 show the information gain values calculated for 10, 20 and 30 clusters produced using K-Means for each of the three preprocessing techniques. Note that in the experiments reported here, we only used data in the training set which contains 6,600 records (please refer to section "Data and Preprocessing"). Also note that we ran K-Means 20 times on each of the preprocessed datasets and reported information gain measures as the average among these 20 runs.

Overall, the polynomial fitting preprocessing technique produces the best clustering results by far regardless of the number of clusters produced. We can note the trend that as the number of clusters increases from 10 to 20 to 30, information gain increases as well. This is expected as a larger number of clusters increases the probability of producing clusters with higher purity.

Results using the averaging preprocessing technique rank a distant second while results for the increasing/decreasing segment counting preprocessing are the poorest. This clearly indicates that the latter preprocessing technique fails to accurately represent the semantics of the data which could be attributed to the fact that the number of increasing/decreasing segments is not indicative of the

digit being spoken. It could more easily correlate with the length of time it took a speaker to utter the digit, for instance. However, both techniques follow the same pattern of improving quality as number of clusters increases for the same reasons outlined earlier.

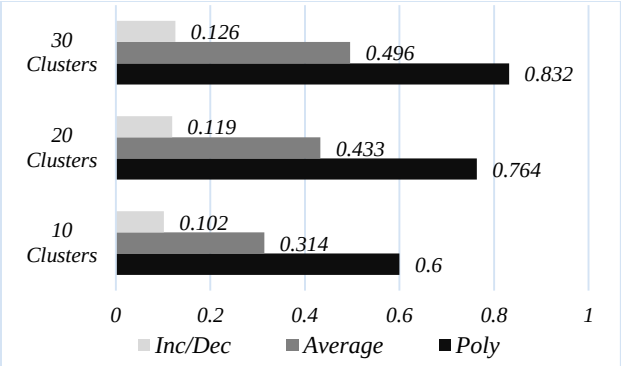

Figure 3: *Info gain results using various preprocessing techniques computed for 10, 20 and 30 clusters.*

Conclusion and Future Direction

Overall, our study of the three preprocessing techniques indicates that the proposed technique of polynomial fitting performs most effectively when using K-Means and evaluating resulting clusters using information again. The averaging technique performs better than the increasing/decreasing segment counting technique which is the poorest by far. Clustering results from all techniques seem to improve as the number of produced clusters increases.

While this study has reported results using K-Means, a future direction for this work would be to produce results using different clustering algorithms including hierarchical and density-based ones. We also envision additional future work focusing on comparing results from the proposed polynomial fitting technique with other preprocessing techniques in the literature especially dynamic time warping (Muller 2007).

In order to further validate the resulting clusters, we plan to use them to classify previously unseen test records. This may be accomplished by classifying records in the test set (refer to section "Data and Preprocessing") based on the nearest centroid. The most frequent class label in the chosen cluster may then be assigned as the predicted class label for the test record. The objective here would be to further validate the use of information gain to evaluate the resulting clusters and consequently support our claims about the superior performance of the proposed polynomial fitting preprocessing technique.

References

Bache, K., and Lichman, M. 2013. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. [Online; accessed October 13, 2014 from https://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit]

Beigi, H. 2011. *Fundamentals of Speaker Recognition*. New York: Springer.

Garrett, J. T.; Heller, K. W.; Fowler, L. P.; Alberto, P. A.; Fredrick, L. D.; and O'Rourke, C. M. 2011. Using Speech Recognition Software to Increase Writing Fluency for Individuals with Physical Disabilities. *Journal of Special Education Technology* 26 (1): 25-41.

Hammami, N., and Bedda, M. 2010. Improved Tree model for Arabic Speech Recognition. In *Proceedings of the third International Conference on Computer Science and Information Technology*, 521-526. Beijing, China: IEEE Press.

Hammami, N., and Sellam, M. 2009. Tree distribution classifier for automatic spoken Arabic digit recognition. In *Proceedings of the International Conference for Internet Technology and Secured Transactions*, 1-4. London, UK: IEEE Press.

Muller, M. 2007. Dynamic Time Warping. In *Information Retrieval for Music and Motion*, 69-84. Berlin, Germany: Springer-Verlag.

Quinlan, J. R. 1986. Induction of Decision Trees. *Journal of Machine Learning* 1(1): 81-106.

Sarfraz, H.; Hussain, S.; Bokhari, R.; Raza, A. A.; Ullah, I.; Sarfraz, Z.; Pervez, S.; Mustafa, A.; Javed, I.; and Parveen, R. 2010. Large vocabulary continuous speech recognition for Urdu. In *Proceedings of the Eighth International Conference on Frontiers of Information Technology*. Islamabad, Pakistan: ACM Press.

Shrivastava, K.; Singhal, N.; Das, P. K.; and Nair, B. S. 2013. A Speech Recognition Client-Server Model for Control of Multiple Robots. In *Proceedings of the International Conference on Advances in Robotics*, 1-6. Pune, India: ACM Press.

Tan, P. N.; Steinbach, M.; and Kumar, V. 2005. *Introduction to Data Mining*. Boston: Pearson Addison Wesley.

Vazhenina, D.; Kipyatkova, I.; Markov, K.; and Karpov, A. 2012. State-of-the-art speech recognition technologies for Russian language. In *Proceedings of the Joint International Conference on Human-Centered Computer Environments*, 59-63. Aizu, Japan: ACM Press.

Zahorian, S. A.; Zimmer, A. M.; and Meng, F. 2002. Vowel Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, 973-976. Denver, Colorado.

Zheng, F.; Zhang, G.; and Song, Z. 2001. Comparison of Different Implementations of MFCC. *Journal of Computer Science and Technology* 16(6): 582-589.