Unsupervised learning of acoustic events using dynamic time warping and hierarchical K-means++ clustering

Joerg Schmalenstroeer[1], *Markus Bartek*[2], *Reinhold Haeb-Umbach*[3]

Department of Communications Engineering, University of Paderborn, Germany
schmalen@nt.uni-paderborn.de[1], bartek@mail.uni-paderborn.de[2], haeb@nt.uni-paderborn.de[3]

Abstract

In this paper we propose to jointly consider Segmental Dynamic Time Warping and distance clustering for the unsupervised learning of acoustic events. As a result, the computational complexity increases only linearly with the dababase size compared to a quadratic increase in a sequential setup, where all pairwise SDTW distances between segments are computed prior to clustering. Further, we discuss options for seed value selection for clustering and show that drawing seeds with a probability proportional to the distance from the already drawn seeds, known as K-means++ clustering, results in a significantly higher probability of finding representatives of each of the underlying classes, compared to the commonly used draws from a uniform distribution. Experiments are performed on an acoustic event classification and an isolated digit recognition task, where on the latter the final word accuracy approaches that of supervised training.

Index Terms: unsupervised, clustering, acoustic events

1. Introduction

Acoustic event classification is concerned with determining the identity of sounds and their temporal position in audio signals. This information can be used to draw conclusions about the physical environment or the activity that has produced the sound. Such an acoustic scene analysis may be used for automatic highlight detection in videos, for surveillance tasks, ambient assistant living devices, or smart environments in general. As it depends very much on the application which acoustic events are relevant, it is desirable to learn the events in an unsupervised fashion to avoid the need for the costly collection of application-specific labeled training data.

There are at least two major approaches to unsupervised pattern discovery in audio or speech data. The first employs machine learning techniques, such as non-negative matrix factorization (NMF), to find recurrent basic building blocks, which in the case of speech could be related to phonemes [1]. A critical issue is how to capture the temporal correlation of the data, as NMF per se provides no handle for this. While several extensions of NMF have been proposed in this direction, the second approach is more directly addressing the sequence character of the input data: Segmental dynamic time warping (SDTW) has been proposed to discover recurring speech patterns in audio streams. Using the matrix of pairwise distances between segments computed by SDTW, graph clustering techniques are applied to identify recurrent temporal patterns [2]. The segment-wise time warping method from [3] extends the set of allowed warping paths for an elaborate time-series matching at the expense of an increased computational complexity.

While unsupervised pattern discovery by SDTW was shown to achieve high cluster purity, a major drawback are its high computational costs, which renders the methods quickly unfeasible for large data sets. Methods have therefore been developed to address this issue. In [4] an approach called "unbounded dynamic time warping" was proposed which utilizes so-called synchronization points, i.e. alignment points which restrict the search space and thus avoid the exhaustive computation of the complete distance matrix. Additionally, local restrictions, that do not allow for strict insertions or deletions, are applied to further restrict the number of warp paths to be evaluated. An overview of sequence matching approaches can be found in [5].

In the following we will present an approach that greatly reduces the computational costs of SDTW-based pattern discovery. We argue that the computation of the complete matrix of pairwise distances between all N_{tot} audio segments is unnecessary, as we will never carry out a full search to obtain the optimal clustering, because this is NP hard. However, if suboptimal, greedy algorithms are applied, it is advantageous to consider the steps of distance computation by SDTW and clustering together and compute only those distances that are actually required for clustering. This will bring down the computational complexity from an order of $\mathcal{O}(N_{\text{tot}}^2)$ operations to $\mathcal{O}(K \cdot N_{\text{tot}})$, where K is the number of seed values of the clustering, which is much smaller than N_{tot}. Further we will investigate the use of the K-means++ algorithm for the selection of the seed values, which will be shown to result in a particularly small number of seed values required to achieve a high probability of choosing a seed value from each of the different acoustic event classes.

The paper is organized as follows. We will first briefly describe SDTW in Section 2 and then discuss our clustering approach, in particular the choice of seed values and the interaction with SDTW. In the experiments section we quantify the computational savings and show that a clever choice of seed values leads to improved cluster purity, both on a speech and an acoustic event database. The paper is finished with conclusions drawn in Section 5.

2. Segmental Dynamic Time Warping

We consider the unsupervised learning of acoustic patterns. Let $\mathcal{E}$ be the set of acoustic event classes, whose cardinality $E = |\mathcal{E}|$, i.e. the number of different event classes, is assumed to be known. We are given a database $\chi = \{x_1, \ldots, x_{N_{\text{tot}}}\}$ of N_{tot} recordings of acoustic events x_i with N events per category, i.e.: $N_{\text{tot}} = N \cdot E$.

Our goal is to partition the database into E clusters such that each cluster ideally contains only the samples of one event class. In principle this can be solved by computing a similarity measure among all pairs of segments and conduct hierarchical clustering until the target number of E clusters has been ob-

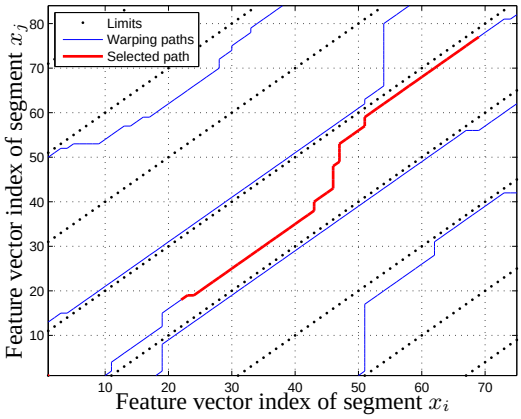

Figure 1: Illustration of segmental dynamic time warping

tained. Dynamic time warping (DTW) has been shown to be a most successful approach to obtain a single distance value describing the similarity of two time series of feature vectors. Note that the individual acoustic events can be of very different length, even if from the same class. Consider for example a ringing tone, whose length may vary between less than a second and several seconds. For this reason it is necessary to extend DTW towards the segmental DTW (SDTW) to find recurring subsequences, as proposed by [2]. SDTW consists of two main components: a local alignment procedure which produces multiple warp paths, and a path trimming method which retains only the lower distortion regions of an alignment path.

In Fig. 1 an example is depicted. The black dotted lines indicate the constraints introduced to restrict the allowable shapes that a warp path can take. Within each diagonal region DTW is applied to find an alignment path (blue lines in Fig. 1). Finally, a path refinement is carried out on each local alignment path to identify the length-constrained minimum average (LCMA) distortion fragment (red line in Fig. 1). The LCMA is that subsequence of the alignment path of a certain minimum length which achieves a minimum value of the distance measure, see [2] for details. This so found value of the distance measure is then taken as the distance $d(x_i, x_j)$ between the two acoustic events x_i and x_j. In [4], the normalized inner product has been proposed as local distance measure between two feature vectors. This is in line with our own findings, where it consistently delivered better final clustering results than the Euclidian or Mahalanobis distance.

A problem of SDTW, that has limited its applicability, is its large computational complexity. The distance matrix $\mathbf{D} = \{d(x_i, x_j)\}$, $1 \leq i, j \leq N_{\text{tot}}$ is input to the subsequent clustering stage. Depending on the clustering method chosen, the availability of the complete distance matrix is however not necessary, as will be explained next.

3. Clustering Approach

With the assumption that the SDTW distance $d(x_i, x_j)$ is smaller if the two acoustic events x_i and x_j are from the same event class than if they are from different classes, an appropriate clustering criterion function is the minimization of the sum of the squared distances between the elements within a cluster. While finding the optimal partitioning is NP-hard, Lloyd has proposed an iterative algorithm, commonly referred to as K-means, that finds a local optimum [6].

3.1. Choice of Seed Values

In K-means it is common practice to choose the K initial centers uniformly at random from the set of data points. Arthur and Vassilvitskii have shown that an alternative initialization leads to a better average value of the criterion function [7]. They have proven that their so-called K-means++ method is $\mathcal{O}(\log K)$-competitive, i.e. the expected value of the criterion function after initialization is no more than a factor worse than the optimum value, where this factor is given by $8(\log K + 2)$ [7].

The K-means++ initialization works as follows:

1. Set $k = 1$. Choose the first seed value c_k uniformly at random from the set χ of acoustic events.
2. Compute the SDTW distances $d(c_k, x_j)$ between the chosen event c_k and all other acoustic events x_j, $j = 1, \ldots, N_{\text{tot}}$ and store the distances in the vector $\boldsymbol{d}_{\min}$.
3. Increment k and choose the next seed value $c_k \in \chi$ with probability proportional to the distances in $\boldsymbol{d}_{\min}$.
4. Compute the SDTW distances between c_k and all other acoustic events and replace an entry in the minimum distance vector $\boldsymbol{d}_{\min}$ if the computed distance is smaller than the stored value.
5. Goto 3. until K centers are drawn.

The idea behind this kind of initialization is to prevent elements of χ to be drawn which are very close to the set of already drawn seed values. On the other hand, although insignificant outliers in χ may have a great distance to the set of previously drawn elements, the probability to draw one of them is small, since the overall number of outliers is per definition small.

The reason for trying to avoid seed values which are close to each other is that we want to find representatives of as many different classes as possible. It is advantageous to have all acoustic event classes represented in the set $\mathcal{C} = \{c_1, \ldots, c_K\}$ of seed values to achieve a high cluster purity in the subsequent cluster refinement steps.

3.2. Probability of Seed Values from all Classes

As there are E different acoustic event classes in the set χ, at least $K{=}E$ seed values are necessary to have at least one seed value from each class. The probability of indeed drawing samples from E different classes with just E draws, is, however, pretty low.

Let the random variable D_k denote the number of different classes after k draws. Obviously, $P(D_1{=}1) = 1$, since the first draw will always yield a representative of a yet unseen class. For the second draw this first sample has to be removed leaving $N-1$ samples from the already chosen event class and $N(E-1)$ from the other classes. If the probability of a sample to be drawn is the same for all samples, i.e. the common initialization of k-means with draws from a uniform probability, the probability of having drawn samples from E different classes after E draws is easily computed to be

$$P(D_E{=}E) = \prod_{k=1}^{E-1} \frac{N(E-k)}{k(N-1) + N(E-k)}. \qquad (1)$$

For $E = 13$ and $N = 300$ we obtain $P(D_{13}{=}13) = 2.1{\cdot}10^{-5}$.

The K-means++ initialization significantly raises this probability. The probability of observing L different classes after k draws can be recursively computed by

$$\begin{aligned} P(D_k{=}L) = {} & P_{\text{new}}(k) P(D_{k-1}{=}L{-}1) \\ & + P_{\text{seen}}(k) P(D_{k-1}{=}L) \end{aligned} \qquad (2)$$

with $P_{\text{new}}(k)$ being the probability of drawing a segment from a yet unseen class in the k-th draw and $P_{\text{seen}}(k)$ the probability of drawing a segment from an already seen class. For the computation of these probabilities we make the assumption that the distance between samples from the same class is the same for all samples and is given by d_0, while the distance between samples from different classes is given by d_1. Then we find

$$P_{\text{new}}(k) = \frac{N[E-(L-1)]d_1}{N[E-(L-1)]d_1 + [N(L-1)-(k-1)]d_0}. \quad (3)$$

The numerator is the number of samples in the $[E-(L-1)]$ yet unseen classes, multiplied by the inter-class distance d_1, while the second term of the denominator is the number of samples remaining in the already seen classes, multiplied by the intra-class distance d_0. Likewise the probability of drawing an already seen class is given by

$$P_{\text{seen}}(k) = \frac{[NL-(k-1)]d_0}{[N(E-L)]d_1 + [NL-(k-1)]d_0}. \quad (4)$$

With the initialization $P(D_1{=}L) = 1$ for $L = 1$ and zero for $L = 2,\ldots,E$, the probability $P(D_E{=}E)$ can be readily computed. The expected value $\mathrm{E}[D_k]$ of the number of different classes seen after k draws is given by

$$\mathrm{E}[D_k] = \sum_{L=1}^{E} L \cdot P(D_k{=}L). \quad (5)$$

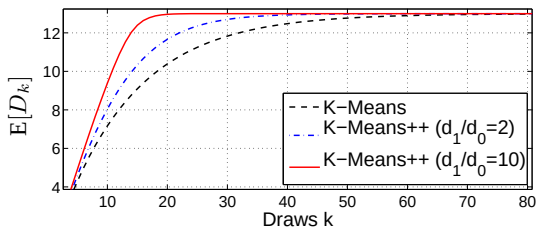

Figure 2: Comparison between K-means variants (number of different classes: $E = 13$)

Figure 2 shows the expected value of different classes seen after k draws as a function of the number of draws k. For K-means++ we show results for an inter-to-intra class distance ratio of $d_1/d_0 = 2$ and 10, whereas K-means draws seed values from a uniform distribution, which corresponds to $d_1/d_0 = 1$. Obviously, K-means++ requires fewer seed points to achieve a certain value of $\mathrm{E}[D_K]$ than uniform initialization.

While the analysis in this section has shown that K-means++ delivers superior seed values compared to uniform sampling, it is unable to predict the degree of performance advantage on true data due to the coarse approximation of a constant inter-to-intra class distance d_1/d_0 which was necessary to arrive at the analytic results.

3.3. Hierarchical Clustering

The overall computational complexity is dominated by the SDTW distance computation. From the description of the K-means++ initialization it can be seen that the required number of SDTW distance computations for seed selection equals $K \cdot N_{\text{tot}}$. The number of seed values, K, should therefore be chosen as small as possible. On the other hand each event class should be represented in the seed values to achieve a final high cluster purity, see later experimental results. Therefore, K should be chosen larger than E to achieve a value of $\mathrm{E}[D_K]$ that is close to E, even for K-means++.

Subsequently, hierarchical clustering is applied to bring the number of clusters down to E. Let $\mathcal{C}_k$ denote the set of seed values merged to the k-th cluster during hierarchical clustering. Initially, all $\mathcal{C}_k$, $k = 1,\ldots,K$, contain a single element, c_k. As a measure of the average inter-cluster distance $\widetilde{d}(C_k, C_l)$, $1 \le k,l \le K$ we define:

$$\widetilde{d}(C_k, C_l) = \frac{2}{|\mathcal{C}_k| + |\mathcal{C}_l|} \sum_{c_i \in \mathcal{C}_k} \sum_{c_j \in \mathcal{C}_l} d(c_i, c_j) \quad (6)$$

Note that all required SDTW distances have already been computed during the K-means++ seed selection. Then the two clusters are merged which exhibit the smallest inter-cluster distance. After each cluster merging the cluster assignment can be iteratively updated by iterating between assigning samples to the closest cluster and recomputing the cluster representative.

4. Experimental Results

We performed experiments on two different databases. The first is the TIDIGIT [8] database where we used the subset of isolated digits only, consisting of 11 classes with 2464 and 2486 digits in the train and test sets, respectively. This database is used to verify that our approach to unsupervised digit recognition is competitive. The second database is the CHIL acoustic event detection database [9]. It contains 3014 non-stationary acoustic events from 13 classes recorded in meeting rooms, e.g. foot steps, knocking, ringing.

We used Mel-frequency cepstral coefficients (MFCCs) as features which were obtained by the ETSI advanced front-end. The feature vectors consisted of 13 MFCCs, an autocorrelation feature and their first- and second-order derivatives.

4.1. Performance Measures

To measure the performance of the proposed approach we define the cluster purity to be the minimal percentage of incorrectly labeled acoustic events among all bijective (one-to-one) mappings $M(\mathcal{C} \leftrightarrow \mathcal{E})$ between the E clusters $\mathcal{C}$ and the E acoustic event classes $\mathcal{E}$. The requirement of bijectivity results in a more stringent definition of cluster purity than the one often used, e.g. in [10].

Please note that for E classes faculty of E mappings exist. We significantly reduced the average computational complexity of searching for the optimal mapping by using a recursive tree traversal algorithm. To this end, we made a first guess of a mapping and subsequently searched only those tree branches which could result in lower error rates than the best mapping found so far.

In the experiments we will also present the normalized mutual information (NMI) as an additional performance measure (see [10] for details).

4.2. Clustering Results

As the seed selection contains a random component we conducted multiple clustering experiments. For each row in the subsequent Tables 1 and 2 we carried out a total of 3000 clustering experiments and report the average, as well as the best and worst cluster purity.

With the joint clustering and SDTW approach proposed here the computational effort is reduced by a factor of $[N_{\text{tot}}(N_{\text{tot}}-1)/2]/[K \cdot N_{\text{tot}}]$ compared to the full SDTW distance computation required in a sequential SDTW and clustering approach. For 11 draws and $N_{\text{tot}} = 2464$ segments of the digit database this amounts to a factor of 112, while a factor

of 116 of computational savings results with the 13 draws and 3014 segments of the acoustic events database. Since the computational complexity is proportional to the number of draws, both for K-means and K-means++, the column showing the number of draws is an indicator of the computational effort of the corresponding experiment.

Method	Draws	Cluster purity [%] Min	Max	Avg.	NMI [%] Avg.
K-means	11	24.72	73.34	48.79	48.75
K-means++	11	22.44	76.50	51.96	51.75
K-means, HC	22	32.22	81.13	58.33	58.28
K-means++, HC	22	32.31	83.60	60.54	60.90
K-means, HC	33	38.39	85.96	63.02	63.16
K-means++, HC	33	42.09	85.35	64.60	65.45
K-means, HC	165	60.15	94.28	82.80	80.36
K-means++, HC	165	63.27	95.41	83.25	81.86
K-means, HC	330	75.37	95.58	89.22	85.91
K-means++, HC	330	69.28	95.86	89.54	86.74

Table 1: Clustering results on speech data

Tables 1 and 2 show the clustering results for speech data and acoustic event data, respectively. As can be seen, cluster purity on the speech database is much higher than on the acoustic events database. Actually, the latter is much more challenging, since the recordings are reverberated, the number of different classes is higher (13 vs. 11), and the average duration of the segments is smaller. Additionally, acoustic events have the disadvantage that samples from the same event class may have very different lengths.

Method	Draws	Cluster purity [%] Min	Max	Avg.	NMI [%] Avg.
k-means	13	12.71	48.31	28.68	32.47
k-means++	13	15.46	48.27	32.12	34.76
k-means, HC	26	14.40	50.20	32.76	37.20
k-means++, HC	26	18.35	53.55	35.75	39.21
k-means, HC	39	16.82	51.96	35.27	39.82
k-means++, HC	39	21.13	52.89	38.13	42.02
k-means, HC	195	29.06	58.99	46.39	52.06
k-means++, HC	195	31.95	63.60	47.84	55.35
k-means, HC	390	39.35	61.68	49.11	57.00
k-means++, HC	390	37.23	58.93	48.90	58.42

Table 2: Clustering results on acoustic event data

In both tables it can be observed that the cluster purity increases with the number of draws. This, however, comes at the price of an increased computational effort which rises proportional with the number of draws. No significant further increase in purity was, however, observed if the the number of seed values was increased beyond the largest value in the tables.

As can be seen in the tables, K-means++ in general outperforms K-means, both in terms of the cluster purity and the normalized mutual information performance measure. On the speech database it can be observerd that K-means catches up to some extent if there are many more seed values than classes ($K \gg E$). The computational complexity of both approaches is similar. K-means requires no distance calculations for drawing the K centers, but it needs $K \cdot N_{\text{tot}}$ distance calculations for assigning the N_{tot} segments to the K clusters. K-means++ requires $K \cdot N_{\text{tot}}$ distances for drawing the K centers and in parallel assigns the N_{tot} segments without additional calculations.

4.3. Speech Recognition Experiment

To gain an intuitively perhaps easier to interpret assessment of the clustering quality than the cluster purity measure, we used two exemplary clustering results to train digit Hidden Markov Models (HMMs) and tested them on the test data set of the digit database (Tab. 3, "First training"). Additionally, the trained HMMs were used to recognize the training data and the recognition results were in turn used to improve the clustering, which was then input to a second training (Tab. 3, "Iter. training"). This increased the cluster purity of the training data and the word accuracy on the test data. As can be seen, unsupervised training is able to approach the word accuracy of supervised training.

Cluster purity [%] First training	Iter. training	Word accuracy [%] First training	Iter. training
80.2	83.61	83.75	84.11
93.50	97.89	99.60	99.72
Perfect clustering (supervised)		99.88	

Table 3: Automatic speech recognition using clustering results

5. Conclusions

We have presented a new approach for learning acoustic event classes in an unsupervised manner. It utilizes segmental dynamic time warping to compare feature vector time-series and in parallel clusters the segments into categories. The approach has the advantage that its computational complexity increases only linearly with the number of utterances and thus enables the clustering of large data sets. Cluster seed value selection via K-means++ requires particularly few seed values to capture representatives of all classes, which eventually pays off in improved cluster purity.

6. References

[1] V. Stouten, K. Demuynck, and H. Van hamme, "Discovering phone patterns in spoken utterances by non-negative matrix factorization," *Signal Processing Letters, IEEE*, vol. 15, pp. 131 –134, 2008.

[2] A. Park and J. Glass, "Unsupervised pattern discovery in speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 186 –197, 2008.

[3] M. Zhou and M. H. Wong, "A segment-wise time warping method for time scaling searching," *Inf. Sci.*, vol. 173, pp. 227–254, June 2005.

[4] X. Anguera, R. Macrae, and N. Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[5] T. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164 – 181, 2011.

[6] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129 – 137, Mar. 1982.

[7] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proc. ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[8] R. Leonard, "A database for speaker independent digit recognition," in *Proc. ICASSP*, 1984.

[9] A. Temko, D. Macho, C. Nadeu, and C. Segura. (2005) CHIL - acoustic event detection. [Online]. Available: http://chil.server.de/

[10] C. Manning, *An introduction to information retrieval*. Cambridge University Press, 2008.