Event Segmentation/Classification

Zachary Kane
Department of Electrical and Computer Engineering, Temple University
zachary.kane@temple.edu

Introduction: The goal of this assignment is to utilize machine learning techniques to parse raw time-series data, perform segmentation to find events in the data and perform sequence decoding (seq2seq) by assigning each event a value of 0-4. Training annotated data is provided which supplied start-times, end-times, and a class value for each file in addition to the raw time-series data

Upon observing the raw information, it is found that the duration for each class pulse follows a uniform distribution. Given that uniform distributions are maximum entropy, little assumptions can be exploited in regards to duration. Similarly, given that the pulses are uniformly distributed with similar pulse shapes, there is little gain in Fourier analysis. The information is encoded in the amplitudes. Each class is provided some initial amplitude pulse, not consistently between files. As more instances of the class' event occur, the amplitudes increase. This results in an important conclusion: the class is dependent on the current observed amplitude of the event and previous occurrences of each class. For this reason, approaches like HMM will fall short as they fail to consider all previous states. Using these observations, two methods will be utilized: a hybrid model (K-Means + LSTM) and a DNN (autoencoder + LSTM).

K-Means + LSTM: The first model utilizes K-means for performing segmentation and an LSTM to perform seq2seq decoding. The data is normalized and smoothed. Each file is then passed through K-means, with K=2. The goal is to convert the signal to a binary form (zeros/ones). The algorithm assumes that one centroid will be representative of the event amplitudes while the other centroid will be representative of the non-event amplitudes (DC offset), given that the signal is relatively sparse. A simple empirical threshold is then used to fine-tune these clusters: if the value is above the lowest centroid plus half a standard deviation, then the point is assigned as event class, otherwise the data point is assigned non-event class. The additional thresholding is necessary to account for small amplitude pulses. The first derivative is applied to the binary signal to provide positive impulses (start times) and negative impulses (end times) and are extracted. The seq2seq decoding is done using an LSTM. The raw data must be quantized first since LSTMs require one-hot encodings. The LSTM contains an encoder step, a context vector and a decoding step optimized with the teacher-forcing principle. The sequence of amplitudes is decoded to a sequence of classes. The LSTM has the advantage of long-term memory so that it may remember previous amplitudes from earlier in the sequence. The output sequence is applied to the Baum-Welsh algorithm for comparing the decoded sequence to the true sequence. A ROC curve is computed using the output statistics of Baum-Welsh.

Autoencoder + LSTM: This approach is similar to the above method, however the K-means algorithm for segmentation is replaced with an autoencoder. Unlike the traditional use-case, the autoencoder is used to map a normalized input to a binary signal representing events/non-events. The autoencoder featured an encoder, a latent space and a decoder. The rest of the algorithm remained the same: using the first derivative to extract timestamps, quantization, applying the LSMT, and evaluating using Baum-Welsh.

LSTM				
Config	***Quantized***	***OutputDimension***	***# Feats***	***Accuracy***
0	100	32	17156	93.62
1	50	32	10756	92.03
2	25	32	7556	91.9
3	100	16	7556	92.08
4	100	8	3524	91.62
5	200	32	7556	85.92

Table 1. LSTM Optimization over feature space

Results: Before evaluation, optimization is performed to ensure the best performance for both the LSTM and the autoencoder. By using different amounts of

Autoencoder			
Config	***Parameter***	***# Feats***	***Accuracy***
0	MSE (Low Compression	263617	84.54
1	BCE (Low Compression)	263617	86.34
2	BCE (High Compression	17156	82.45

Table 2. Autoencoder Optimization over feature space

quantization and internal nodes, the number of features in the LSTM are modified, as shown in Table 1. The best configuration is found to use 100 as the quantization value and an internal node structure of 32. Based on these results, the autoencoder is tuned by varying the amount of compression to the feature space and the cost function, shown in Table 2. Generally, BCE with low compression performed the best, at the expense of many more features. Once optimized, results for the training and dev sets are reported, shown in Table 3. Generally, the K-Means approach proved superior to the autoencoder approach regarding accuracy and F1 score, while the autoencoder received higher AUCs. However, these results are not significant with any confidence above 50%, thus K-Means cannot be determined as a superior algorithm. When examining false-positive rates, there is a much larger discrepancy which was found to be significant up to 80%. In this way, K-Means can be considered superior only when using this metric. In terms of complexity, K-Means is much simpler and does not require supervised training like the autoencoder, making K-Means more attractive. The ROC curves are shown to follow the general trend of descending linear lines, with the autoencoder having a slightly higher curve, Fig 1. However, it is apparent from these curves that both segmentors perform similarly.

System	***DataSet***	***Accuracy***	***F1***	***AUC***	***FP Rate***
Auotencoder + LSTM	Train	86.33	0.55	0.49	16.93
	Dev	86.34	0.55	0.39	17.16
	Eval				
KMeans + LSTM	Train	87.90	0.58	0.43	8.53
	Dev	87.67	0.58	0.35	10.46
	Eval				

Table 3. Summary of metrics

Conclusions: This assignment investigated segmenting a time-series signal and performing sequence decoding. A K-means + LSTM approach is used as a hybrid model followed by a purely DNN method using an autoencoder + LSTM. Many of the metrics resulted in insignificant differences, expect for false-positive rates. In conjunction with its low-complexity, the K-Means approach with LSTM appears to be the better performing method based on the Dev and Train results.

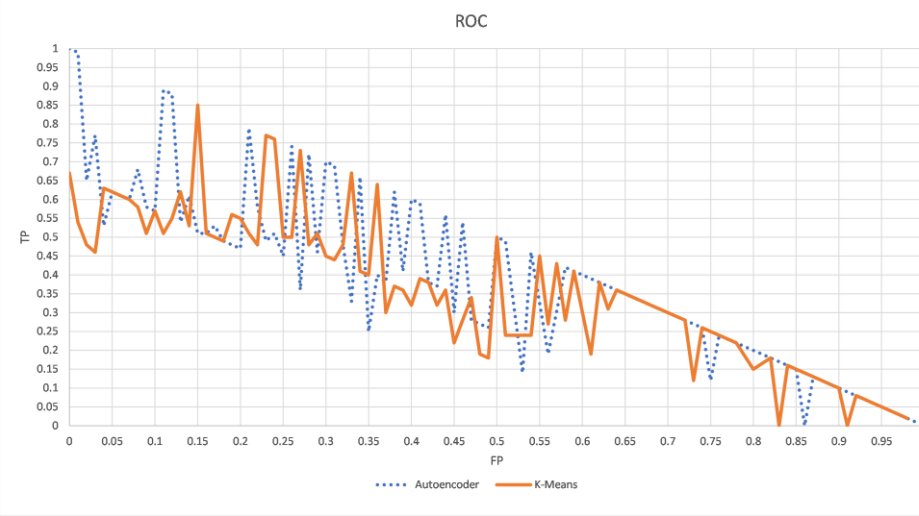

Fig 1. ROC Curves