Matching Multifrequency Clinical Time Series

Yi Wei
Machine Learning Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
ywei1@andrew.cmu.edu

Karen (Lujie) Chen
Heinz College
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
karenchen@cmu.edu

Gilles Clermont
Critical Care Medicine
University of Pittsburgh
3550 Terrace Street
Pittsburgh, PA 15261
clermontg@ccm.upmc.edu

Artur W. Dubrawski
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
awd@cs.cmu.edu

Abstract

In this article we consider the problem of matching time sequences in the MIMIC II[1] database. Unlike other time series similarity matching problems, our task is to match time sequences of observations made at different frequencies that have been obtained from the same subject. For each time series in high-frequency, our goal is to find its low-frequency counterpart. Heart rate of each patient in the ICU is recorded by both the patient monitor automatically and by nurses manually. Series of heart rate recorded by nurses(low-frequcency clinical data) is widely applied to analysis and detection of diseases, while the high-frequency numerical data monitored by bedside monitors do not have the information needed to link it directly with clinical data. We want to match those anonymous numerical data with clinical data for future research. We studied various metrics of time series similarity and proposed two efficient metrics with high accuracies. To evaluate the performance of various metrics, synthesized unmatched pairs, along with provided matched pairs of clinical and numerical datasets, compose the training and testing set. Cross-validation is conducted to evaluate the metrics. Experiments of matching multifrequency series are conducted on the testing set. According to our experimental results, accuracy of detecting true matching is 58.80%. Higher accuracy is expected when a more efficient matching algorithm is implemented.

1 Movitation

Time series are of growing importance in many real-world applications such as stock market quotations, population series, weather forecasts, etc. In medicine, analysis of clinical time series arises as an important research topic. Time series are sequences of real numbers in chronological order. Blood pressure, heart rate, etc are typical examples of clinical time series. Query search in a database of time series is an information retrieval problem of finding the qualified time series which have similar pattern as the query. In a database of clinical time series, the query might be "Find patients who had a similar heart rate pattern as patient S's heart rate during his treatment in the ICU".

The task of our project is to match high-frequency numerical data with low-frequency clinical data in the MIMIC II Clinical Database. The MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) Databases contain physiologic signals and vital signs time series captured from patient monitors,

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

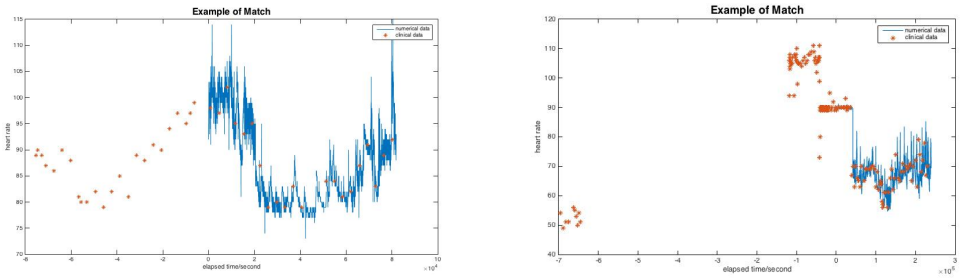

Figure 1: Example of matched pairs

and comprehensive clinical data obtained from hospital medical information systems, for tens of thousands of Intensive Care Unit (ICU) patients [1]. Heart rate of patients in ICU was recorded by both patient monitors automatically and by nurses manually. Monitors recorded value of heart rate per second or per minute, while nurses went to the ICU and wrote down numbers shown on the monitor each 10-60 minutes. Thus for each patient in the ICU, there are two sequences of heart rate, respectively in high-frequency measure and low-frequency measure. Clinical data(low-frequency measure) in MIMIC II is widely used for analysis and research, however, numerical data(high-frequency measure) is discarded at present. Doctors and researchers at UPMC(University of Pittsburgh Medical Center) hold the opinion that research on disease prevention and cure would benefit from inclusion of high-frequency numerical data. Since those numerical data was commonly omitted, information about the patient and the ICU identifier was not saved. The task of our project is to not only find the correct clinical matching for each numerical time series, but also find the time offset between two sequences within a pair. There are 46571 clinical sequences and 15957 numerical sequences in the database, besides 5266 matched pairs already provided as a training set. Figure 1 shows examples of matched pairs.

Our project includes several challenges beyond standard similarity matching problems: 1. Numerical and clinical data are sampled at different frequency, which gives rise to difficulty of indexing. 2. Even matched pairs do not guarantee that two sequences are of the same length. Length here means the time span, not the number of data points in the sequence. Therefore, our system should be tolerant of queries of arbitrary length 3. Nurses might not record the information at the exact time shown in the database. So numerical and clinical data are not subject to an exact temporal alignment, and thus we must design a noise-tolerant metric and matching algorithm.

2 Related Work

There are many methods to measure the similarity of two time sequences, such as Euclidean distance in a multidimensional space[2]. Since data cannot be indexed directly because of its high dimensionality, low dimensional features are extracted from time series and represented as vectors in a multidimensional space. One of commonly exploited transformations to extract features is DFT(Discrete Fourier Transform). First few parameters of DFT[3] are leveraged as the feature of a signal. According to Parseval's theorem, Euclidean distance of two signals in time domain is the same as their Euclidean distance in frequency domain. Therefore, index by utilizing DFT may cause false alarms, however it guarantees no false dismissals. Other transformations applied to signals in order to reduce the dimension of feature vectors include Discrete Wavelet Transform (DWT), Karhunen-Loeve (KL) transform, etc. [4] demonstrates that Euclidean distance is preserved in the Haar transformed domain and no false dismissal will occur.

Similarity matching in MIMIC II dataset can be considered in a statistical view. Each low-frequency data can be regarded as a sample from the group of high-frequency data within the 15 min window centered at low-frequency timestamp. Therefore, numerical data and its clinical matching can be sliced into segments, and in each segment, there is a clinical data and a group of numerical data. Z-score of low frequency measure w.r.t high frequency group and p-value can be calculated for each segment. P-value estimates the probability that a pair composed of low-frequency data and the high-frequency dataset are realizations of the same physical process. Multiple p values can be calculated for each pair of clinical series and numerical series using various tests of significance. [5]

compared several ways of combining p-values from individual statistical tests. We use some of the methods in [5] for reference to metrics of time series similarity.

3 Research on Metrics

Popular methods of transforming signals to low dimensional space do not fit our task directly. In the MIMIC II database, data at a lot of timestamps is missing, and also time interval between two consecutive low-frequency measures is not fixed, ranging from 10 minutes to 60 minutes. If we wanted to exploit DFT or DWT to transform signals onto feature space, we would have to recover data and make sure that they share the same time interval between two consecutive measures. Fabricating data is not the correct direction we should follow, therefore, we explored other directions.

This section introduces metrics utilized to measure similarity between time series and establishes experiments to evaluate their performance. In the database there are 5266 matched pairs, while 4150 of them are valid pairs. Invalid pairs include those time series for which heart rate is all missing and those pairs where numerical signal and clinical signal actually do not share same timestamps. Unmatched pairs were synthesized in three ways. Unmatched pairs, along with matched ones, compose the training dataset for the purpose of evaluating performance of various metrics.

3.1 Synthesis of Unmatched Pairs

In the dataset, clinical matching are provided to 4150 numerical signals, and thus those numerical signals with any clinical signals other than their matching can form unmatched pairs. The first method to synthesize unmatched pairs is randomly picking mismatching for numerical time series in the dataset of matched pairs. Adding a random time offset to timestamps of clinical sequence in each matched pair is another way of synthesizing unmatched pairs. Similarly, we can also randomly shuffle timestamps of matched pairs in order to make them unmatched.

One of the three methods is randomly chosen when generating the same amount of unmatched pairs as matched ones. After collecting those matched and unmatched pairs, experiments of evaluating various metrics were established on this training dataset. Value of metrics was calculated for each pair and ROC(Receiver Operating Characteristic) curves were plotted for each metric. By comparing ROC curves for different metrics we can find the suitable metrics for similarity matching.

3.2 Measuring Similarity at Matched Timestamps

We can obtain two longest subsequences in a pair sharing the same timestamps, by computing the intersection of timestamps of the two time series. Metrics of measuring similarity at matched timestamps are discussed in this subsection.

There are several correlation coefficients quantifying statistical relationships between two populations. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

Spearman's rank correlation coefficient is a nonparametric measure of statistical dependence between the ranking of two populations. For a population of size n, the n original values X_i, Y_i are converted to ranks r_{X_i}, r_{Y_i}, then spearman's coefficient is calculated as [6]:

$$\rho = \rho_{r_X, r_Y} = \frac{cov(r_X, r_Y)}{\sigma_{r_Y} \sigma_{r_Y}} \tag{2}$$

Kendall rank correlation coefficient is also a measure of rank correlation. It is defined as [7]:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)} \tag{3}$$

where C is the number of concordant pairs and D is the number of discordant pairs. A pair is concordant if the ranks for both elements agree, that is, $(X_i - X_j)(Y_i - Y_j) > 0$. A pair is discordant if the ranks for elements disagree, that is, $(X_i - X_j)(Y_i - Y_j) < 0$.

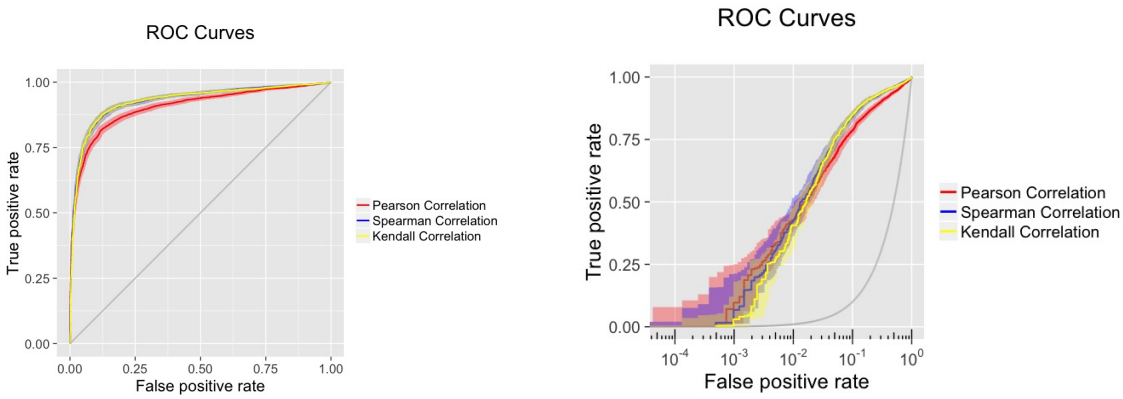

Figure 2: **Left:** ROCs with confidence intervals for correlation coefficients; **Right:** false positive rate in log scale

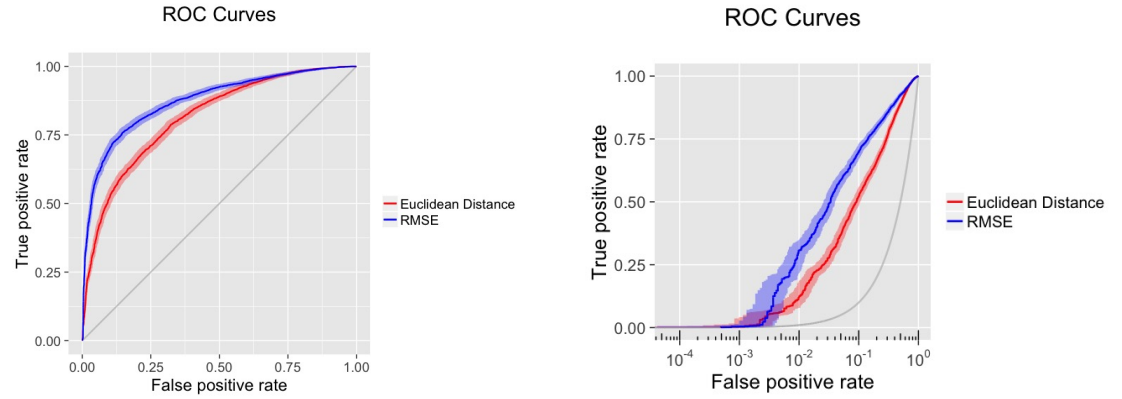

Figure 3: **Left:** ROCs with confidence intervals for RMSE and Euclidean distance; **Right:** false positive rate in log scale

"Matched" and "Unmatched" can be regarded as two classes, and then the matching problem will be a binary classification problem. Value of metrics can be leveraged as the score to measure the probability of a pair being classified as the positive(matched) label. ROC curves with confidence intervals for the three metrics are shown in Figure 2. Confidence intervals, shown at 95% level, are calculated by using the Wilson's score method[8]. The corresponding AUC's are: Pearson: 0.9043, Spearman: 0.9276, Kendall: 0.9269.

Root-mean-square error (RMSE) and Euclidean distance can also be exploited as metrics. Figure 3 shows the ROC curves with confidence intervals. AUC for Euclidean Distance is 0.8134 and RMSE is 0.8736. We can observe from Figure 3 that RMSE performs better than Euclidean distance. For the reason that length of matched pairs in the dataset are diverse, RMSE, which is a regularized error measure, definitely has better performance than Euclidean distance, the metric relying on the length of sequence.

3.3 Measuring Similarity by Hypothesis Testing

There are several relevant metrics related to hypothesis testing. At each low frequency timestamp, find high-frequency measure within 15-minute window centered at low frequency timestamp. We assume that high-frequency data within 15-minute window follows normal distribution. Therefore we can compute p-value from the z score of low-frequency data w.r.t high-frequency data points. Tail probability (denoted by tp) from the empirical distribution can also be utilized as a metric. Multiple aggregate p-value and tail probability methods include: 1. $\sum_{i=1}^{n} log(p_i)$; 2. $\frac{\sum_{i=1}^{n} log(p_i)}{n}$; 3. $\sum_{i=1}^{n} p_i$; 4. $\frac{\sum_{i=1}^{n} p_i}{n}$; 5. $\sum_{i=1}^{n} log(tp_i)$; 6. $\frac{\sum_{i=1}^{n} log(tp_i)}{n}$; 7. $\sum_{i=1}^{n} tp_i$; 8. $\frac{\sum_{i=1}^{n} tp_i}{n}$; 9. Ac-

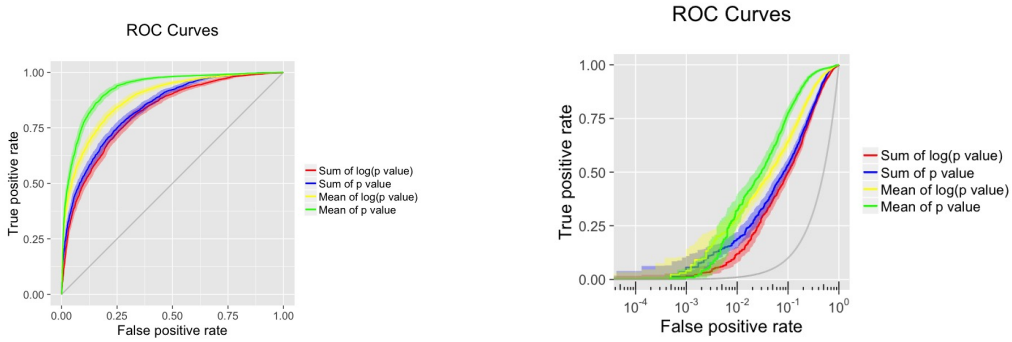

Figure 4: ROC curves for metrics: 1. $\sum_{i=1}^{n} log(p_i)$; 2. $\frac{\sum_{i=1}^{n} log(p_i)}{n}$; 3. $\sum_{i=1}^{n} p_i$; 4. $\frac{\sum_{i=1}^{n} p_i}{n}$. The corresponding AUC's are 0.8168, 0.8801, 0.8362, 0.9236. **Left:** ROCs with confidence intervals for aggregate p-values methods; **Right:** false positive rate in log scale

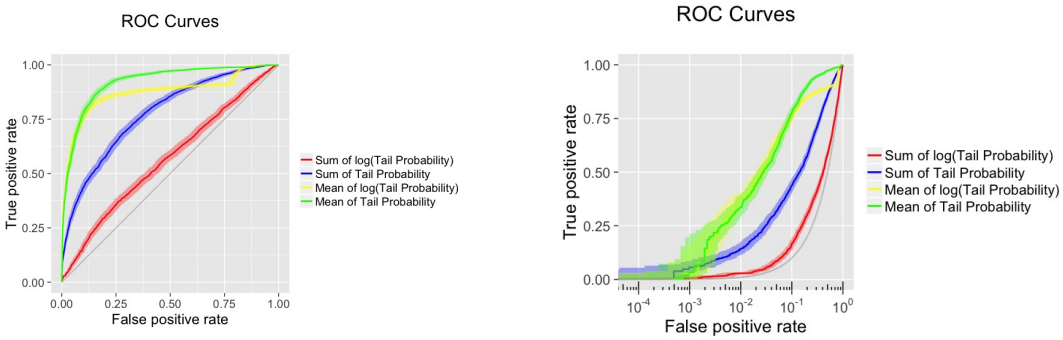

Figure 5: ROC curves for metrics: 5. $\sum_{i=1}^{n} log(tp_i)$; 6. $\frac{\sum_{i=1}^{n} log(tp_i)}{n}$; 7. $\sum_{i=1}^{n} tp_i$; 8. $\frac{\sum_{i=1}^{n} tp_i}{n}$. The corresponding AUC's are 0.5648, 0.8727, 0.7752, 0.9197. **Left:** ROCs with confidence intervals for aggregate tail probabilities methods; **Right:** false positive rate in log scale

ceptance rate of hypothesis testing by p-value: $\frac{\sum_{i=1}^{n} I_{p_i>0.05}}{n}$; 10. Acceptance rate of hypothesis testing by tail probability: $\frac{\sum_{i=1}^{n} I_{tp_i>0.05}}{n}$; 11. Tail strength measure[9] of p-values: the test statistic compares the ordered p-values to the expected moments of the Uniform(0,1) distribution[5]: $TS_p = \frac{1}{m}\sum_{i=1}^{n} \frac{i-p_{(i)}(n+1)}{i}$, where $p_{(i)}$'s are ordered p-values 12. Tail strength measure of tail probabilities: $TS_{tp} = \frac{1}{m}\sum_{i=1}^{n} \frac{i-tp_{(i)}(n+1)}{i}$, where $tp_{(i)}$'s are ordered tail probabilities. ROC curves for those metrics are shown in Figures 4, 5, 6,7 and their corresponding AUC's are calculated as well.

3.4 Combination of Metrics

Among metrics introduced in Section 3.2 and 3.3, there are 12 having AUC larger than 0.85. They are RMSE, Pearson coefficient, Spearman coefficient, Kendall coefficient, $\frac{\sum_{i=1}^{n} log(p_i)}{n}$, $\frac{\sum_{i=1}^{n} p_i}{n}$, acceptance rate of hypothesis testing by p-value, tail strength measure of p-values, $\frac{\sum_{i=1}^{n} log(tp_i)}{n}$, $\frac{\sum_{i=1}^{n} tp_i}{n}$, acceptance rate of hypothesis testing by tail probability, tail strength measure of tail probability. Since those metrics interpret time series similarity in different perspectives, we should not simply select the metric with highest AUC and then discard the others. Here we would like to introduce linear composite models to maximize the benefit from combining those various metrics.

PCA(Principal Component Analysis) is applied to scores calculated by various metrics in order to eliminate the redundancy between metrics. Before we apply PCA, scores should be normalized. Since PCA projects raw scores onto directions which maximize the variance, if scores were not normalized, only metric with large variance would contribute. We can observe from the covariance matrix that there exists strong positive and negative correlation among metrics. At the same time, the fast decay of eigenvalues in Figure 8 also demonstrates the redundancy of metrics. Therefore, we can use the

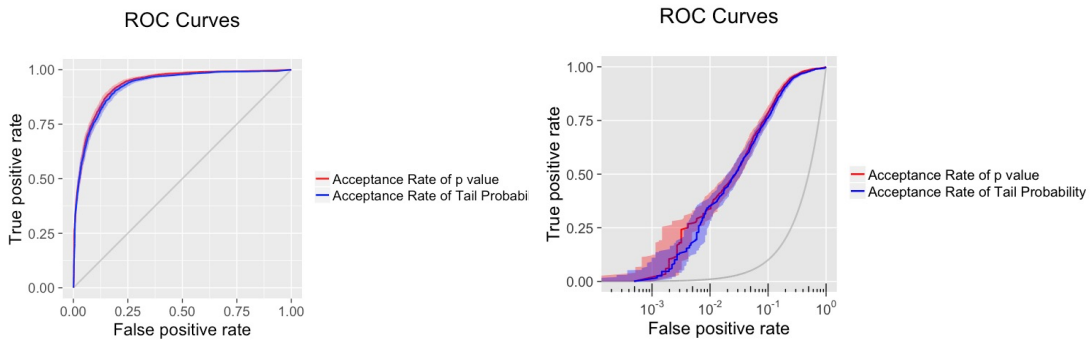

Figure 6: ROC curves for metrics: 9. Acceptance rate of hypothesis testing by p-value: $\frac{\sum_{i=1}^{n} I_{p_i>0.05}}{n}$; 10. Acceptance rate of hypothesis testing by tail probability: $\frac{\sum_{i=1}^{n} I_{tp_i>0.05}}{n}$. The corresponding AUC's are 0.9313 and 0.9239. **Left:** ROCs with confidence intervals for acceptance rate methods; **Right:** false positive rate in log scale

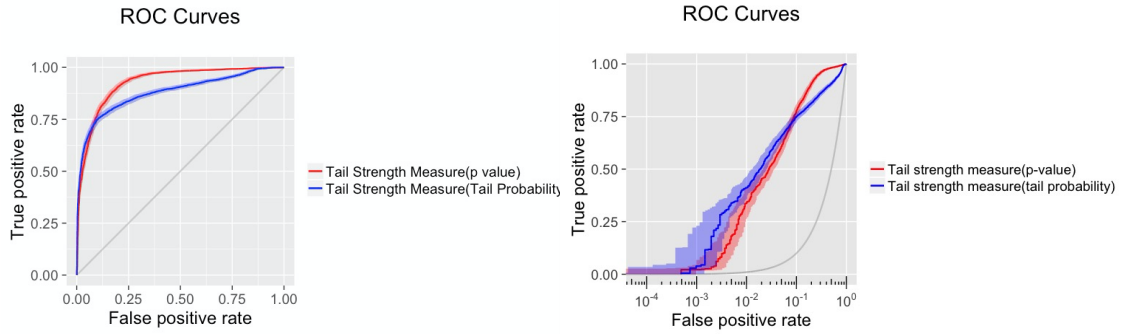

Figure 7: ROC curves for metrics: 11. Tail strength measure of p-values; 12. Tail strength measure of tail probabilities. The corresponding AUC's are 0.9252 and 0.8786. **Left:** ROCs with confidence intervals for tail strength measure; **Right:** false positive rate in log scale

largest component as our final metric of similarity. The eigenvector w.r.t. the largest eigenvalue is exploited as the weights of metrics in their linear combination as shown in Table 1. For the purpose of evaluating the performance of the metric from PCA, cross-validation with 10 folds are applied. And ROC curve is plotted in Figure 9 and $AUC = 0.9741$. Both ROCs and AUC demonstrate good performance of this metric. Table 1 shows the weight of each metric in the eigenvector w.r.t the largest eigenvalue. Some of these weights are negative, for the reason that smaller scores of those metrics represent more similarity. For instance, RMSE measures the distance between two sequences, and thus there is a strong negative correlation between similarity and RMSE.

Another way to derive the final metric is applying Logistic Regression on scores. Scores of different metrics are viewed as features of pairs, "matched" as the positive label and "unmatched" as the negative. Cross-validation with 10 folds are applied to the Logistic Regression model. Cross-validated accuracy of 10 folds is {0.93057247, 0.9001218 , 0.9317905 , 0.92570037, 0.94762485,

metric	weight	metric	weight	metric	weight
rmse	-0.2281	mean_log_p	0.2205	mean_log_tp	0.2363
pearson	0.2635	mean_p	0.3270	mean_tp	0.3303
spearman	0.2730	acr_p	0.3291	acr_tp	0.3300
kendall	0.2770	tail_strength_p	-0.3107	tail_strength_tp	-0.3052

Table 1: Weight of metrics in main component from PCA

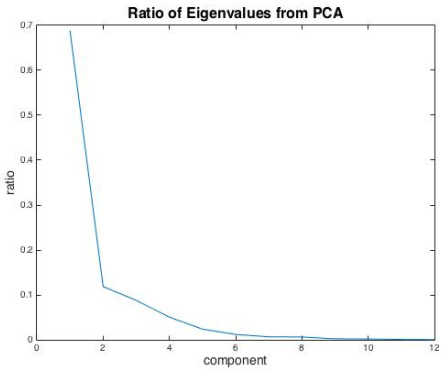

Figure 8: Ratio of eigenvalues from PCA

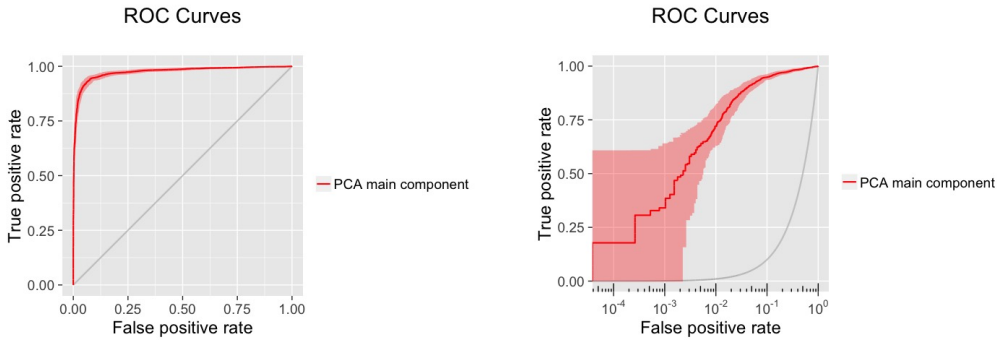

Figure 9: **Left:** ROCs with confidence intervals for PCA main component; **Right:** false positive rate in log scale

0.93292683, 0.93780488, 0.93406593, 0.92551893, 0.92063492}. Figure 10 plots the ROC curves for the results of cross validation. $AUC = 0.9733$.

4 Testing on the Training Dataset

4.1 Setup of Experiments

Experiments of matching time series by using metrics in Section 3.4 were established on the training datasets. The matching algorithm utilized in this section is linear scanning, that is, for each numerical

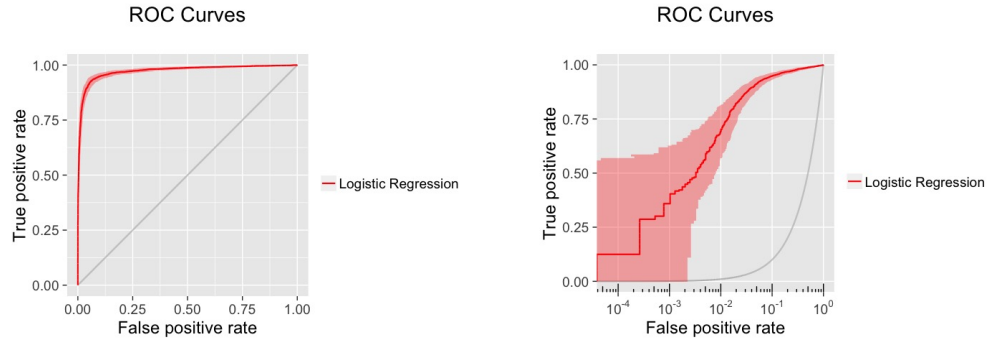

Figure 10: **Left:** ROCs with confidence intervals for Logistic Regression; **Right:** false positive rate in log scale

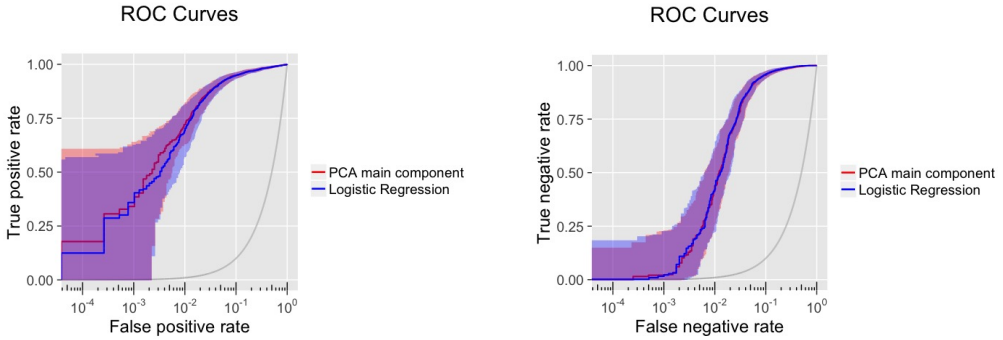

Figure 11: **Left:** ROC for Logistic Regression and PCA, TPR VS. FPR; **Right:** ROC for Logistic Regression and PCA, TNR VS. FNR. We can conclude from the graphs above that compared with detecting true negatives, our metrics have better performance at detecting true positives. Since negatives could be classified as the positives incorrectly, therefore, in the matching stage(Section 4.1), rather than find the candidate with the highest score, we instead output a list of candidate matches sorted by their scores in the descending order.

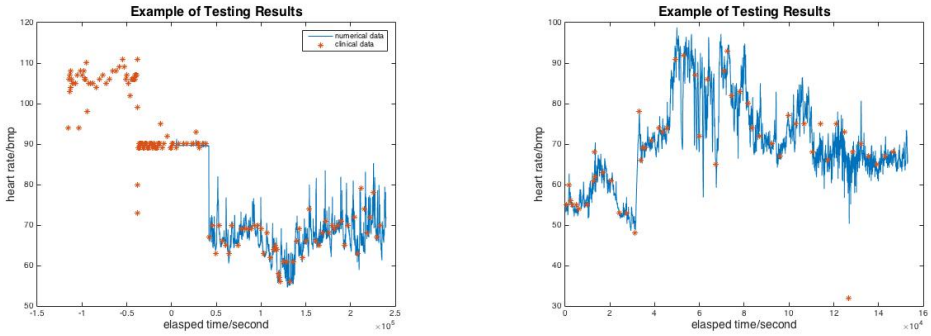

Figure 12: Examples of correct matching. **Left:** using metric from PCA, score = 3.502, **right:** using metric from LR, score = 7.859.

signal, firstly downsampling itself and then scanning all clinical time series and every possible time offset. For each clinical time series, find the optimal offset, and compute its score. After linear scan of all clinical time series, we output a list of top candidates sorted by their pairwise scores in decreasing order. Ideally, true matching for numerical time series are expected to gain high rank. Most high frequency sequences in the database have $10^5 \sim 10^7$ data points and there are 46571 clinical sequences in the dataset, so it is time consuming to search all clinical signals and every possible time offset within a pair. Therefore, our current experiment was set up in a smaller scope. We shrunk the number of candidates by randomly picking 100 clinical sequences for each numerical time series and then included the true match into the candidates list. At the same time, to prune the search space, we assume that two sequences within a pair align roughly either at the head or the tail. Thus after making heads or tails of two sequences align, we only search time offsets within ± 8 hours.

4.2 Results of Experiments

The method of measuring accuracy is introduced in this subsection. It is the percentage of pairs where rank of the true matching is among top 10, and we denote it as r_{top}. When we use the main component from PCA as our metric, it turns out that $r_{top} = 58.80\%$. When metric trained from Logistic Regression is applied, $r_{top} = 58.59\%$. Reasons for failure of detecting true matching are discussed in Section 5.

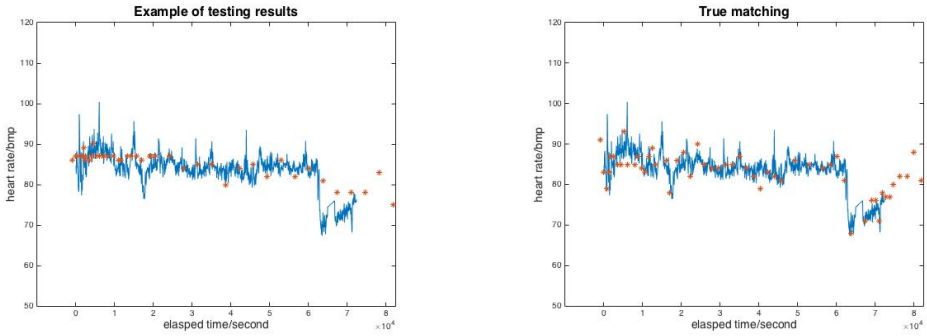

Figure 13: Comparision of incorrect matching and true matching. **Left:** incorrect matching using metric from LR, score = 4.969, **right:** true matching, score = 4.462.

5 Discussion

One of the reason why true matching pairs failed to be selected into candidates is that we shrunk the searching space for the purpose of reducing the computational burden, and in many cases the true time offset is out of our search scope. Thus, an efficient and not too time-consuming searching algorithm is desired.

Due to the low signal variability of some pairs such as the pair shown in Figure 13, matching is expected to be hard. For a time sequence with low entropy or variation, we can extract very little information from it. Especially if a low frequency sequence has low variation, it can fit high-frequcency sequences of many shapes, in which case false positives will easily occur. The left graphs in Figure 13 and 14 plot incorrect matchings which rank first among all the candidates. In the pairs shown in two graphs, the true matchings are out of the top 10 lists. We can observe that both the incorrect and true matching have similar shapes. Therefore, metrics optimizing the trade-off between exact matching and tolerance of noise still need to be further investigated.

6 Summary

We reviewed multiple types of metrics of similarity and identified a few that show good performance at detecting matching patterns between multifrequency time series. In experiments with the proposed composite models, the probability of detecting true matches is 59%. Our composite metrics are weighted combinations of metrics computing distance at matched timestamps and metrics measuring the probability of clinical observation following the same distribution of its closest numerical population. The second type of metrics ensure that our metrics are tolerant to noise in temporal alignment in the datasets. Nurses might not record the heart rate at the exact timestamp recorded, and so MIMIC II numerical and clinical data are not subject to an exact temporal alignment.

Our matching pipeline will be exploited further to enable practically reliable alignment of the now disconnected data to benefit further research on prediction and analysis in ICU settings. In the future work, we will improve time efficiency of the linear scanning algorithm. Since our algorithm output a list of matchings with highest scores for each numerical sequence, we need to identify the optimal links in sparse bipartite graphs of entities representing numeric and clinical data obtained from the same cohorts of patients by embedding matching scores as inputs for the assignment problem. Furthermore, in our project, we match multifrequency time series via heart rate, in the future, we could try to enable matching with respect to multiple vital signs simultaneously.

References

[1] https://physionet.org/mimic2/.

[2] Kahveci, Tamer, and Ambuj K. Singh. "Optimizing similarity search for arbitrary length time series queries." IEEE Transactions on Knowledge and Data Engineering 16.4 (2004): 418-433.

[3] Faloutsos, Christos, Mudumbai Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. Vol. 23. No. 2. ACM, 1994.

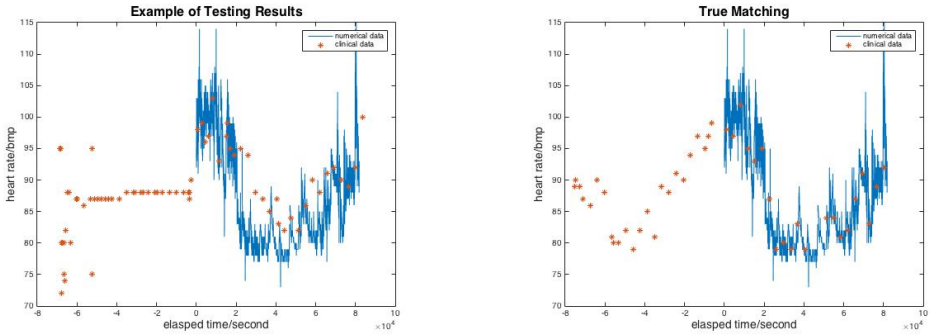

Figure 14: Comparision of incorrect matching and true matching. **Left:** incorrect matching using metric from PCA, score = 1.671, **right:** true matching, score = 0.9647.

[4] Chan, Kin-Pong, and Ada Wai-Chee Fu. "Efficient time series matching by wavelets." Data Engineering, 1999. Proceedings., 15th International Conference on. IEEE, 1999.

[5] Mitchell, Matthew W. "A Comparison of Aggregate P-Value Methods and Multivariate Statistics for Self-Contained Tests of Metabolic Pathway Analysis." PloS one 10.4 (2015): e0125081.

[6] Myers, Jerome L.; Well, Arnold D. (2003). Research Design and Statistical Analysis (2nd ed.). Lawrence Erlbaum. p. 508. ISBN 0-8058-4037-0.

[7] Nelsen, R.B. (2001), "Kendall tau metric", in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4

[8] Wilson, Edwin B. "Probable inference, the law of succession, and statistical inference." Journal of the American Statistical Association 22.158 (1927): 209-212.

[9] Taylor, Jonathan, and Robert Tibshirani. "A tail strength measure for assessing the overall univariate significance in a dataset." Biostatistics 7.2 (2006): 167-181.