QUANTIFYING RELATIONSHIPS BETWEEN TWO TIME SERIES DATA SETS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Arighna Roy

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2016

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

QUANTIFYING RELATIONSHIPS BETWEEN TWO TIME SERIES DATA SETS

By

Arighna Roy

The supervisory committee certifies that this thesis complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Anne Denton
Chair

Dr. Saeed Salem

Curt Doetkott

Approved:

November 17, 2016
Date

Dr. Brian M. Slator
Department Chair

ABSTRACT

One of the popular methods for quantifying the relationship between two time series data sets is canonical correlations; however, it is linear and cannot accommodate more complex scenarios, such as time series data for which distance relationships are best characterized through dynamic time warping. I introduce a nearest-neighbor-overlap method that resolves both problems and allows a reliable determination of significant relationships. The nearest neighbor algorithm also does not depend on the normal distribution of the variables, unlike canonical correlations. Also, it is not sensitive to singularity (when one variable is derivable from another) of the data. I demonstrate that our method substantially outperforms canonical correlation analysis for time series data sets from the UCR repository as well as the environmental data of Red River Valley region.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation through grants PFI-1114363 and IIA-1355466

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Time series, a sequence of data points for consecutive and equally spaced (preferably) time points, is a pervasive form of data almost in every scientific application. With the surge of time series data sets, it has become critically important to identify whether these data sets are related. The relationships can be identified by finding the correlations between pairs of data sets. To take a practical example for the motivation of our research, let's consider agricultural data. The agricultural community has to know which factors affect plant growth. Finding the strength of relationship between weather time series data sets and vegetation time series data sets serves that purpose. In contrast between a multivariate data set and a monthly precipitation time series data set, each observation vector of the multivariate data set is represented by a time series of length 12 (monthly precipitation data) for a specific coordinate in the time series data set. And each column of the multivariate data set is represented by a specific time point(a month for this data) for all the coordinates.

The existing parametric methods of correlation analysis make assumptions which most of the time series data do not conform to. Canonical correlation analysis is based on the concept of finding the linear combination of input and output data points which yields the maximum correlation between them. Tests that were developed for identifying relevant input in multivariate multiple regression can be used to assess the significance of correlations. But the statistical significance tests for multivariate multiple regression such as Wilks' lambda, union–intersection test(Roy's test equivalent), Pillai and Lawley-Hotelling count on the following assumptions [15]; multivariate normality of both the input and the output variables and linear relationship among the input and the output variables. The assumption of multivariate normality is impractical for most time series data sets. For example, the spatial pattern of the precipitation shows skewed frequency distributions [20]. Also, time series data do not always interact with each other with a linear combination of the data points. Furthermore, most of these tests fail when the sample size is smaller than the number of dimensions. High Dimension, Low Sample Size (HDLSS) [6] data is very common for time series data as well as other fields of science.

In this paper, an algorithm is discussed to establish relationships between two time series data sets without making assumptions about the underlying distribution of the data points and pattern of the relationship between the data sets. The instances with similar pattern in the input data set are likely to follow similar pattern in output data set as well, if the two data sets are related. With increasing number of time points (or length of time series), the probability of similar behavior by random chance becomes smaller. I measure the overlap of the nearest neighbors for each instance in both input and output dimensions to determine if there is a significant relationship between them.

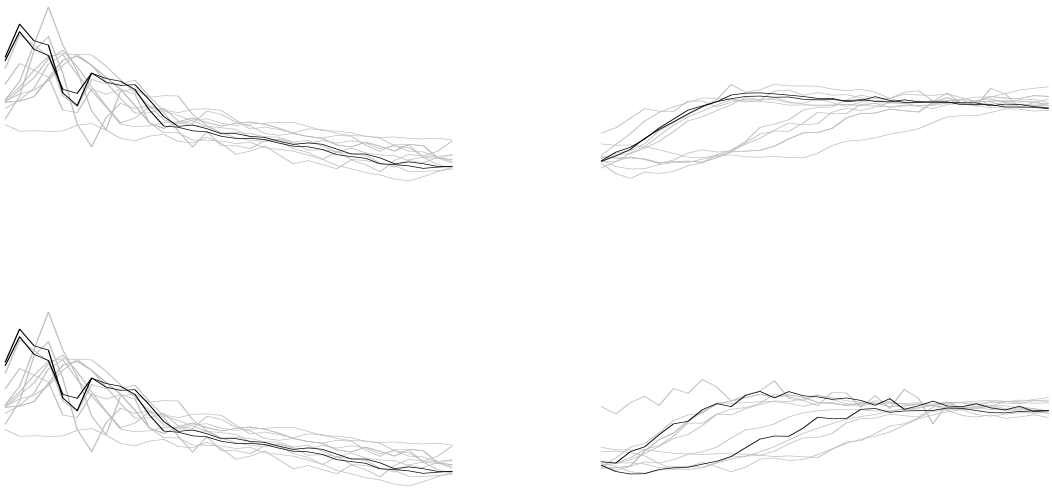

Figure 1.1. Schematic to illustrate the problem: The top left figure shows a set of 100 time series. Top left panel shows one of the time series and its nearest neighbor in black. The top right figure shows the right side of the same observations as the top left one. The same two time series are shown in black.

I have used one single time series data set to illustrate our idea. Each instance of the time series data sets is vertically split into two halves. Suppose there are n instances(time series) in a data set and each of length m. So the each half of the data set will consist of n observations and each observation is of length $m/2$. Now, for one single observation, the probability of its nearest neighbor in the left(first) half of the data set to also be the nearest neighbor in the right(second) half of the data set is normally higher than expected by random chance alone, as they originally

belong to the same time series data set. In contrast, when the rows for the right haves of the instances are randomly shuffled, there is not expected to be a relationship between the data sets anymore.

I have used a figure of four panels below to schematically explain our assumption. The top left panel of figure 1.1 shows the first half of a subset of a time series data set. Two nearest neighbors in that space are highlighted in black. The top right panel shows the second half for the same set of observations. Here, the same nearest neighbors are also highlighted in black. Data points for the second halves of both the time series in right panel are highly probable to be nearest neighbors as well.

In the bottom two panels, I show that the trend is not the same in both halves due to random chance only. In bottom right panel, I highlight the second half of time series with same indexes as the nearest neighbors from the first half. I reshuffle the indexes of the time series only for the right half. The same two nearest neighbors from the top left panel are highlighted in the bottom left panel. Those highlighted time series are not neighbors any more.

I have used multiple time series data sets to test the algorithm as well as checked its performance on agricultural data sets. In first case, I have constructed the left and right data sets by splitting one single data set vertically into two equal haves. Because of the vertical split, the time points for the two data sets do not overlap. In second case, I have used precipitation data as the left data set and vegetation data as the right data set. Here the time points for the data sets overlap. The time points for precipitation data cover the duration of one year. Whereas, the time points for vegetation data cover a subset (Jun-Aug) of the same year.

2. RELATED WORK

Among the related analysis of correlation between two sets of vectors, canonical correlation analysis (CCA) is one of the most robust and useful one. [8] It's a parametric method to analyze the relationship between two sets of vectors. It gives answer to in how many reliable dimensions, the variables are related to each other. Canonical correlations find the optimal combination of input and output variables' coefficients in multiple dimensions and the corresponding magnitude of associations for those combinations. The reliability of a dimension is determined by the corresponding magnitude of association (eigenvalue) for that specific combination of variables in that dimension. And the variables' coefficients are determined from the corresponding eigenvectors. The combination of all the canonical correlations provide a measure of overall correlation between the variables. CCA is derived from multivariate multiple regression and has the similar limitations. Canonical correlation has been used to find correlation between two time series data sets [2]. An extension of CCA is nonlinear canonical correlation analysis using kernel function and neural network. As part of the kernel method, data is non-linearly transformed into a feature space and then CCA is performed on the transformed data into that feature space. For neural network, sigmoid function is applied on input and output variables to establish a relationship between them. [13]

Neural networks are heavily used to find the optimized correlation coeeficients between the data points of two time series data sets [9]. Three feed forward neural networks can be used to perform a nonlinear CCA [10]. The first network maximizes the correlation between the two canonical variates and the other two networks provide the feedback to the variables from the canonical variates. Multilayer Perceptrons can also be used to find the transformed vectors from the given variables [3]. Multiple stacked layers of network have also been used to find the optimized vectors. This method is called as Deep Canonical Correlation [19]. Kernel methods are also used for CCA. Two observation vectors can be transformed into Hilbert spaces to find a feature for which the correlation coefficients are the maximum [1]. Both kernel functions and neural networks can be used to find the nonlinear mapping between the original variables and the transformed data and then linear canonical correlation is performed on the transformed data. Radial basis function and a linear neural network is used here for the mapping task [7].

3. CONCEPTS

3.1. High-Level Overview

I am considering two sets of variables for the same set of sample points and trying to find if the combination of one set of variables is correlated to the combination of the other set of variables. For example, I have the precipitation data and the vegetation data of the same set of geo-spatial points. I have considered the precipitation values at specific coordinates collected at distinct time points as the first set of variables and the vegetation index collected at distinct time points (not necessarily those intervals would coincide) for the same coordinates as the second set of variables. Our task is to find the answer if the precipitation data and the vegetation data are significantly correlated.

I have considered multiple approaches and data sets to portray the consistency of our claim. Different length of dimensions, randomly sampled instances and multiple model parameters have been considered to conclude our argument.

3.2. Assumptions for k-NN Intersection Algorithm

The idea of the algorithm is that if an observation has k nearest neighbors in one vector representation, and the same nearest neighbors in the other, the two vector representations are likely to be related. This likelihood can be calculated by the overlap of the nearest neighbors collectively for all the observations. Suppose, the nearest neighbor of the m^{th} observation of the left data set is the n^{th} observation of the same data set; also the nearest neighbor of the m^{th} observation of the right data set is the n^{th} observation of that data set. Then we count it as one overlap and this overlap is due to the m^{th} observation of both the data sets. Regardless of how many of the neighbors overlap, the probability of the observation can be calculated. A generalization to evaluating the significance based on a combination of all instances is straightforward.

3.3. Basic Concepts

Consider a data set with two sets of variables, $x = (x_1, x_2, ..., x_m)$ and $y = (y_1, y_2, ..., y_n)$ with the length of m and n respectively. Now the i^{th} sample of the data set would be as follows: $(x_{i1}, x_{i2}, ..., x_{im}, y_{i1}, y_{i2}, ..., y_{in})$. Now, suppose x_i is the nearest neighbor of x_j based on a specific distance metric (like Euclidean distance). y_i is expected be the nearest neighbor of y_j as well based on the same distance metric.

3.4. k-NN Intersection Algorithm

In the k-NN Intersection Algorithm I consider neighborhoods consisting of the k_{nn} nearest neighbors to an instance l in both vector spaces:

$$(U_l^a \mid \forall m \in U_l^a, n \notin U_l^a,\ d(\mathbf{x}_l, \mathbf{x}_m) \leq d(\mathbf{x}_l, \mathbf{x}_n) \wedge |U_l^a| = k_{nn})$$

and

$$\Big(U_l^b \mid \forall m \in U_l^b, n \notin U_l^b,\ d(\mathbf{y}_l, \mathbf{y}_m) \leq d(\mathbf{y}_l, \mathbf{y}_n) \wedge |U_l^b| = k_{nn}\Big)$$

For the calculation of the expected overlap of the nearest neighbors of $\mathbf{x}_l$ and $\mathbf{y}_l$, the observation that is characterized by $\mathbf{x}_l$ and $\mathbf{y}_l$ is excluded from the consideration, and the probabilities are calculated over $N-1$ points. Each of those observations has a probability of $k_{nn}/(N-1)$ of being in the subset that's defined by being $\mathbf{x}_l$ and, independently, a probability of $k_{nn}/(N-1)$ of being in the subset that's defined by being $\mathbf{y}_l$. Overall, the expected number of shared neighbors within any pair of unrelated neighborhoods is $\frac{k_{nn}^2}{N-1}$. The number of shared elements of neighborhoods is furthermore aggregated over each of the possible instances l resulting in a total of

$$n_{expect} = E\left(\sum_l |U_l^a \cap U_l^b|\right) = \frac{k_{nn}^2 N}{N-1}$$

if vector attributes **A** and **B** are unrelated. This aggregation is possible because the definition of neighborhoods does not impose any kind of mutual exclusion between sets.

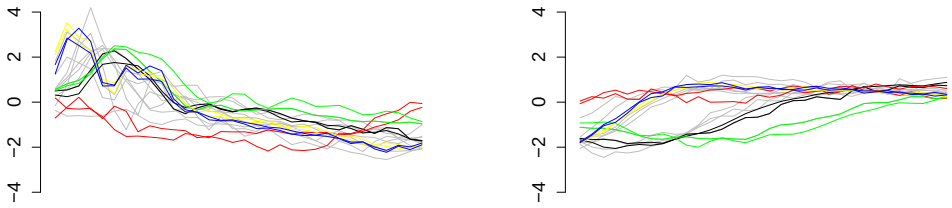

Figure 3.1. Schematic to illustrate the idea of K-NN Intersection algorithm: The top left figure shows ecg200 time series for 32 time points. I have chosen 5 observations and their nearest neighbors. Each of such pairs are of same color. The top right panel shows the same 5 sets of instance pairs as the left figure for a different set of 32 time points.

After the aggregation, the expected values are large enough that the Poisson distribution that is used for comparison can be approximated by a Gaussian distribution. The number of standard deviations above the mean, z is reported, with $z > 2$ considered an indication that the distributions are related.

$$z = \frac{\left|\sum_l |U_l^a \cap U_l^b| - n_{expect}\right|}{\sqrt{n_{expect}}}$$

3.5. Dynamic Time Warping

Analyzing temporal data includes identifying similarity (or distance) between the time series. The most popular and widely used distance measure metric for numeric data is Euclidean distance. But Euclidean distance doesn't consider the shape of the two series among which I need to find the distance. Consider two sine curves of different frequency length but of same total length. The Euclidean distance between these two time series would be quite high which is very unlikely. Dynamic Time Warping (DTW) solves this shortcoming by finding the optimum path between the time series which in turn favours the similarity in shape. [16]

Let us consider two time series A and B of same length as following,

$A = (a_1, a_2, a_3, ...a_n)$

$B = (b_1, b_2, b_3, ...b_n)$

The Euclidean distance between these two time series is, $E_{ij} = \sum_{i=1}^{n}(a_i - b_i)^2$

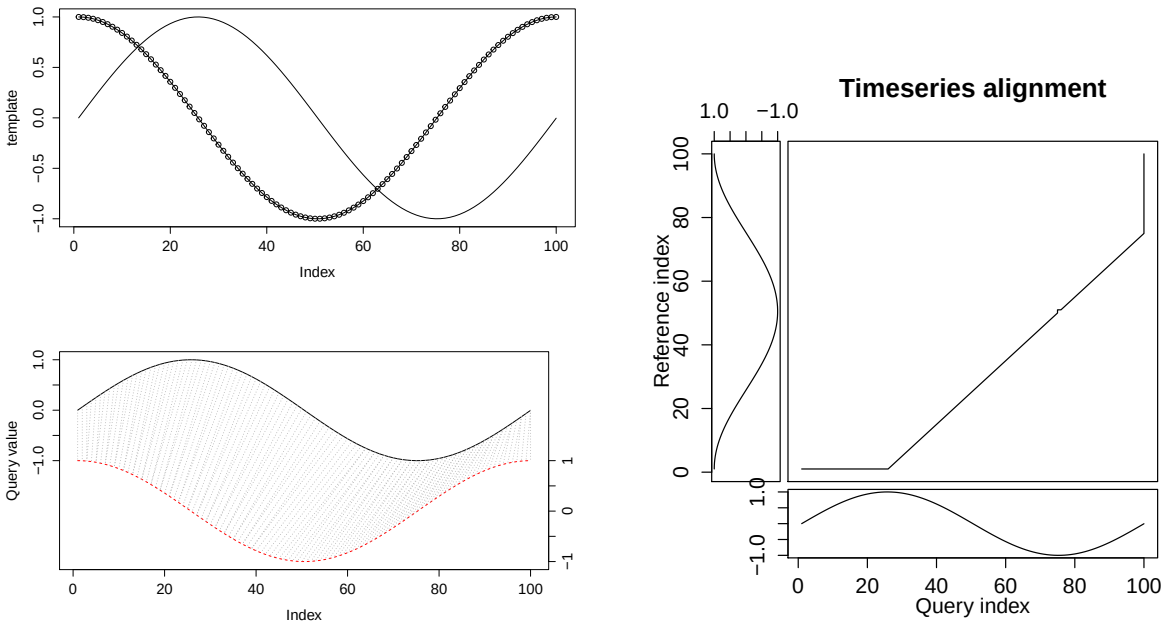

Figure 3.2. Top left: Two sequences that are similar but out of phase. Bottom left: To align the sequences, I construct a warping matrix and search for the optimal warping path. Right: The resulting alignment

I can implement some dynamic programming approach to find an optimum path between two time series. First I have to find the distance matrix D_{i*j} for the two time series. $D_{ij} = (A_i - B_j)^2$ Second, I aim to find a warping path between them. The final distance is calculated to be $F_{ij} = D_{ij} + min(F_{(i-1)(j-1)}, F_{(i-1)j}, F_{i(j-1)})$

I have used dynamic time warping as a measure of distance which makes it very useful for time series with missing values as dynamic time warping works for time series of different length. Dynamic time warping works best while used with $k = 1$ for k-NN Intersection Algorithm.

4. THE PSEUDOCODE

Algorithm 1 K-NN Intersection

procedure ENRICHKNN(X,Y,k) ▷ The correlation between 2 multivariate datasets X and Y

if the length of X and Y are not the same **then**

return with error message

end if

$set1 \leftarrow$ k nearest neighbors of each time series of X

$set2 \leftarrow$ k nearest neighbors of each time series of Y

$n \leftarrow$ number of time series in X or Y

$overlap \leftarrow 0$

for $i \in \{1, ..., n\}$ **do**

$overlap \leftarrow$ overlap+length of intersection between i-th row of set1 and set2

end for

$expected \leftarrow k * k * n/(n-1)$

$result \leftarrow (overlap - expected)/\sqrt{expected}$

return result

end procedure

The procedure enrichKnn() takes exactly 3 arguments; the input data set, the output data set and the value of k. The first two arguments take matrices and the third argument takes an integer. The order of the input and the output data sets doesn't not matter, because correlation is an undirected relationship. Both the data matrices should contain equal number of rows(samples). I find the k nearest neighbors for each of the data sets. The neighbor matrices are n*k matrices. Each row of the neighbor matrices contains k indexes of rows where those k indexed rows are the nearest neighbors for that row. For example, if the 3^{rd} row of the neighbor matrix of data set X is a vector (5,32,43,7,19) then the 5 nearest neighbors for the 3^{rd} row of X are 5^{th}, 32^{nd}, 43^{rd}, 7^{th} and 19^{th} rows. Thereafter I calculate the cumulative overlap in the nearest neighbor indexes for corresponding rows of the neighbor matrices. I also calculated the expected value of the overlap.

Finally significance of relationship is calculated as the $result \leftarrow (overlap - expected)/\sqrt{expected}$. This value is returned as the answer.

5. CANONICAL CORRELATIONS AS COMPARISON ALGORITHM

Let's find the canonical correlations from two sets of time series data. I first build the data matrix from the time series data. Each row of the data matrix is one time series and each column represents a specific time. Suppose, $\mathbf{y}$=(y_1,y_2,...,y_p) and $\mathbf{x}$=(x_1,x_2,...,x_q) are two time series data sets. Each row of $\mathbf{y}$ and $\mathbf{x}$ are time series of length p and q correspondingly. The correlation matrix R for $\mathbf{y}$ and $\mathbf{x}$ can be represented as

$$R = \begin{bmatrix} R_{yy} & R_{yx} \\ R_{xy} & R_{xx} \end{bmatrix}$$

Where,

$$R_{yy} = \begin{bmatrix} r_{y_1y_1} & r_{y_1y_2} & \cdots & r_{y_1y_p} \\ r_{y_1y_2} & r_{y_2y_2} & \cdots & r_{y_2y_p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{y_1y_p} & r_{y_2y_p} & \cdots & r_{y_py_p} \end{bmatrix}$$

R_{yy} is the correlation matrix of $\mathbf{y}$. It's a symmetric matrix. The diagonal elements are ones, because those are the correlations among the same y variable. The off-diagonal elements are the correlations among y variables.

$$R_{xx} = \begin{bmatrix} r_{x_1x_1} & r_{x_1x_2} & \cdots & r_{x_1x_q} \\ r_{x_1x_2} & r_{x_2x_2} & \cdots & r_{x_2x_q} \\ \vdots & \vdots & \ddots & \vdots \\ r_{x_1x_q} & r_{x_2x_q} & \cdots & r_{x_qx_q} \end{bmatrix}$$

R_{xx} is the correlation matrix of $\mathbf{x}$. Similar to R_{yy}, R_{xx} is symmetric and the diagonal elements are ones.

$$R_{yx} = \begin{bmatrix} r_{y_1x_1} & r_{y_1x_2} & \dots & r_{y_1x_q} \\ r_{y_2x_1} & r_{y_2x_2} & \dots & r_{x_2x_q} \\ \vdots & \vdots & \ddots & \vdots \\ r_{y_py_q} & r_{y_px_2} & \dots & r_{y_px_q} \end{bmatrix}$$

R_{yx} is the cross-correlation matrix between $\mathbf{y}$ and $\mathbf{x}$.

$r_{y_ax_b}$ is the correlation between y_a and x_b

R_{yx} is the transpose of R_{xy}

As, both R_{yy} and R_{xx} are symmetric, and R_{yx} is transpose of R_{xy}, R is a symmetric matrix.

Now, the correlation between two variables a and b is defined as, $r_{ab}=\frac{s_{ab}}{\sqrt{s_a} * \sqrt{s_b}}$

where, s_{ab} is the covariance between a and b

s_a is the variance of a and s_b is the variance of b

Now, the covariance between two variables a and b is defined as,

$cov_{a,b}=s_{ab}=\frac{\sum_{i=1}^{n}(a_i-\bar{a})(b_i-\bar{b})}{n-1}$ where n is the number of observations for each a and b

Now, the variance of a variable a is defined as, $var_a=s_a=\frac{\sum_{i=1}^{n}(a_i-\bar{a})}{n}$

Now, $r_1, r_2, ..., r_s$ are the canonical correlations of $\mathbf{y}$ and $\mathbf{x}$. Where $s = \min(p, q)$ and $r_1^2, r_2^2, ..., r_s^2$ are the eigenvalues of $R_{yy}^{-1}R_{yx}R_{xx}^{-1}R_{xy}$. Each of the s canonical correlations provide a linear combination of $\mathbf{x}$s and $\mathbf{y}$s for a multivariate regression line. It can be shown that the eigenvector corresponding to r_1 shows the highest correlation among them.

The significance of the whole set of dependent attributes can be tested using F- approximation given in:

$F = \frac{1-\Lambda_1^{1/t}}{\Lambda_1^{1/t}}\frac{df_2}{df_1}$

where $\Lambda_j = \prod_{i=j}^{s}(1 - r_i^2)$, $s = \min(p, q)$, $df_1 = pq$, $df_2 = wt - \frac{1}{2}pq + 1$, $w = n - \frac{1}{2}(p + q + 3)$, $t = \sqrt{\frac{p^2q^2-4}{p^2+q^2-5}}$

F has an approximate F-distribution with df_1 and df_2 degrees of freedom. I can say that $\mathbf{y}$ and $\mathbf{x}$ are related if $F > F_\alpha$ (The critical value of the F-distribution).

Λ_j is known as Wilk's Lambda and can also be expressed as $\Lambda = \frac{|E|}{|E+H|}$. Where E is the sum of squares error and H is the sum of squares regression. When there is indeed a linear

relationship between $\mathbf{y}$ and $\mathbf{x}$, $|E|$ will much smaller than $|H|$ which in turn will make Λ smaller. The reason I consider only Λ_1 is because Λ_1 considers all the s Canonical Correlations into account; if any of those are large then the value of Λ_1 will become small which in turn will lead to a higher F-value. If the F-value exceeds the critical F-value then it shows significant correlation.

5.1. Limitations of Canonical Correlation

- Firstly, normal distribution of the data; multivariate multiple regression analysis that incorporates discriminant analysis will produce identically the same results as a canonical correlation analysis in terms of dimension reduction analysis. [14] Multiple regression assumes the multivariate normal distribution of both the input and output variables. Although canonical correlations do not limit to multivariate normal data sets, it definitely works better for such data.

- Secondly, Linearity; Canonical Correlation analysis assumes linearity relationships among and between both dependent and independent sets of variables. It cannot capture the nonlinear components of the relationship. This one drastically limits the usefulness of this analysis while dealing with time series data sets which have a non linear relationship between themselves. [5]

- Thirdly, homogeneity of variance; CCA assumes a stable distribution of variance throughout the data which is often not the case in real life dataset.

- Fourthly, singularity; The performance of CCA reduces if there are derived variables in the set of independent variables. Although, this feature can be fulfilled by doing a pre-processing on both the sets of variables separately.

- Fifth, like multivariate regression, CCA perform better for large number of rows.

- Last and the most important problem, its inability to accommodate more sophisticated distance measure techniques like Dynamic Time Warping.

6. EXPERIMENTAL RESULTS

I have implemented the algorithm in R. 'CCA' package is used for canonical correltion analysis. 'FNN' package is used for knn algorithm. The function to find the DTW distance between two time series and the KNN with DTW as distance metric are implemented using basic R scripting. To test our algorithm, I have considered pre-processed time series data sets initially.

Table 6.1. Time series used in the experiments

faceall	Non-Invasive Fetal ECG Thorax1
u Wave Gesture Library X	Star Light Curves
Non-Invasive Fetal ECG Thorax2	wafer

To maintain the robustness of the algorithm, I have iterated the experiment for multiple time series and multiple times for each time series. Later I have repeated the same experiment on agricultural data.

6.1. Time Series Data

To prepare the pre-processed time series data sets for testing our algorithm, I have split the data sets in two equal halves. The first half is considered to be an individual time series data set and the second half is considered to be another. When I test the correlation between these two time series data sets, we expect that they are related. But, if I shuffle the indexes of rows for the right half of the data set then those are not supposed to be correlated.

I have used 6 time series mentioned in table 6.1 from the UCR Time Series Repository [4]. All the results that are reported are aggregated over those 6 time series. To vary the length of the time series, time points are are selected evenly distributed along the time series. For example, a data set consists of time series of length 101 and I want to find the correlation between two halves of a data sets with the length of both the time series to be 10. So I find 20 equally spaced time points (an arithmetic series with the first element to be 1 and the common difference to be 5). And the data points for the first 10 time points are considered to be the first time series and the data points for the rest 10 time points is considered to be second time series.

7. PERFORMANCE OF THE ALGORITHM

The performance of our algorithm is tested with a black-box approach where we test the result of our algorithm, the relationship between two time series data sets, with previously assumed relationships. We do not focus on the internal functionality or the data distribution while testing the performance of the algorithm.

7.1. Terms Related to Performance

I have used Accuracy, Precision, Recall and F1 score as performance metrics. To calculate these metrics, I need to identify each result as one of the below mentioned four categories.

- **True Positive**

  True Positives are those instances which are positive in real and predicted to be positive as well. Let us explain it in the context of our experiment. Suppose one time series data set is split into two halves and those are predicted to be related in our experiment. That is a True Positive case.

- **True Negative**

  True Negatives are those instances which are Negative in real and predicted to be Negative as well. In our experiment, suppose one time series data set is split into two halves and the right half instances are randomly shuffled. Now, if those are predicted to be not related in our experiment then it is a True Negative case.

- **False Positive**

  False Positives are those instances which are Negative in real but predicted to be Positive. In our experiment, suppose one time series data set is split into two halves and the right half instances are randomly shuffled. Now, if those are predicted to be related in our experiment then it is a False Positive case.

- **False Negative**

  False Negatives are those instances which are Positive in real but predicted to be Negative. In our experiment, suppose one time series data set is split into two halves and those are predicted to be not related in our experiment then it is a False Positive case.

- **Error Rate**

  Error rate of an experiment is defined as the ratio of the sum of false positives and false negatives to the total number of cases.

7.2. Dependence on K

I have reported the error rate with respect to the model parameter k, the number of nearest neighbors to be considered, to find the optimal value of k. I have shown the error rate for the values of k=1,2,3,4,5 for varying sample sizes as well as varying number of dimensions.

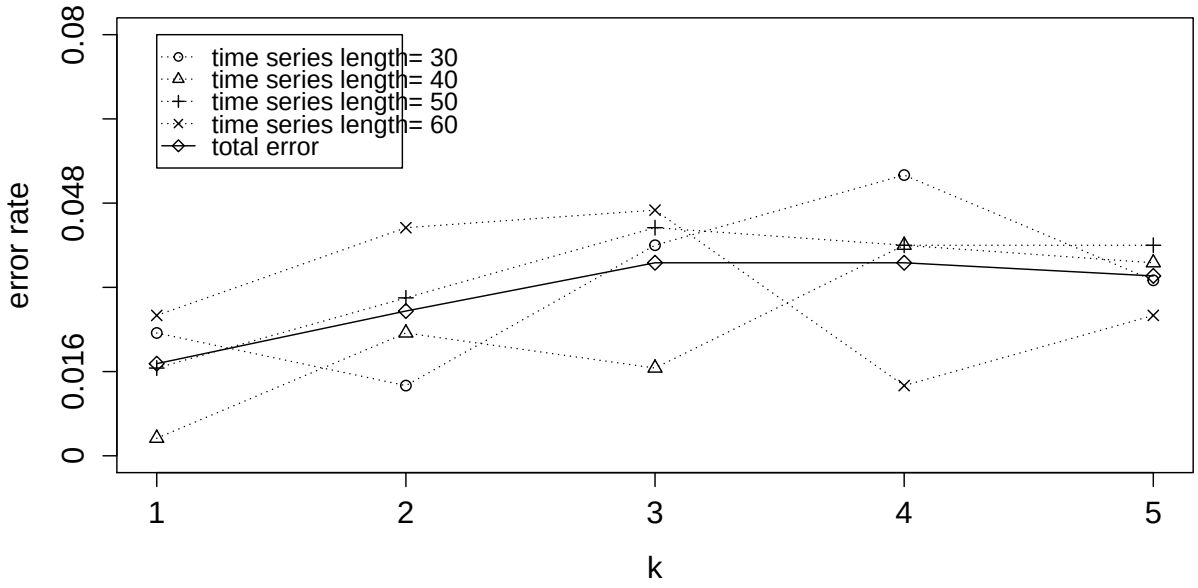

Figure 7.1. Error rate of KNN Intersection algorithm depending on k for different time series length

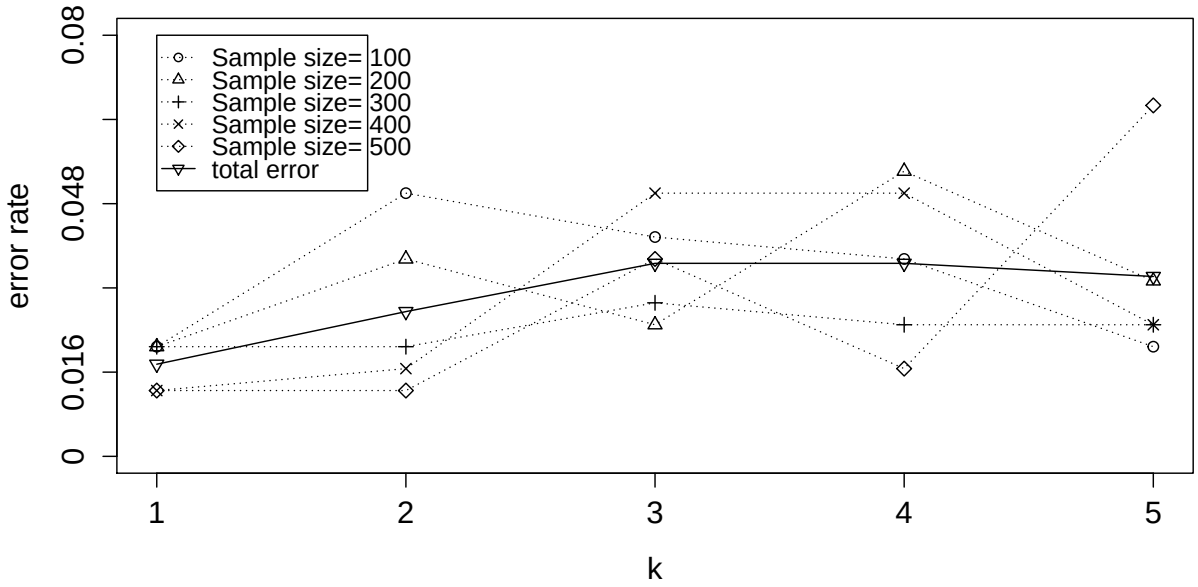

Figure 7.2. Error rate of KNN Intersection algorithm depending on k for different sample size

I intend to find an optimal value of k for which the algorithm gives best performance . Whereas there is no specific trend in error rate for different k with varying sample size or varying time series length, but the error rates, in most cases, are the lowest for k=1. The total error rate almost always increase with k. It changes with k in a similar way for both varying sample size and the length of time the series. So I can safely use $k = 1$ for kNN intersection algorithm for the most accurate result.

Performance of the algorithm was expected to be the best for $k = 1$. The value of n_{expect} follows a quadratic growth with k. But the chances of neighbor overlap by accident also increase with k. Both of the factors increase the error rate with the increase of k.

7.3. Comparative Study of Performance with Canonical Correlation

I have taken randomly indexed samples from the time series data sets to remove any kind of bias towards the order of data collection. This random sampling happens in every iteration.
To evaluate the dependence of the performance of our algorithm on time series length and number of samples, I use the parameters that result in the lowest error rate $k = 1$ for k-NN Intersection Algorithm. I have performed the test on 6 different time series and all the results are shown in the

above figure. The consistency of the trend in the results for all the time series shows the reliability of the test results.

7.4. Dependence on Time Series Length

To test the behavior of our algorithm for varying time series length, I have first repeated our algorithms for different sample sizes and different time series lengths. Now, for every unique value of time series lengths, I have averaged the performance metrics over all the different values of sample size. That makes the result unbiased to any specific sample size. Thereafter the comparative results are reported test the effect of the sample sizes on the performance of the algorithm.

7.4.1. Accuracy

Accuracy of a study is defined as the ratio of correctly classified instances to the total number of instances.

$$Accuracy = \frac{NumberofTruePositives+NumberofTrueNegatives}{Totalnumberofinstances}$$

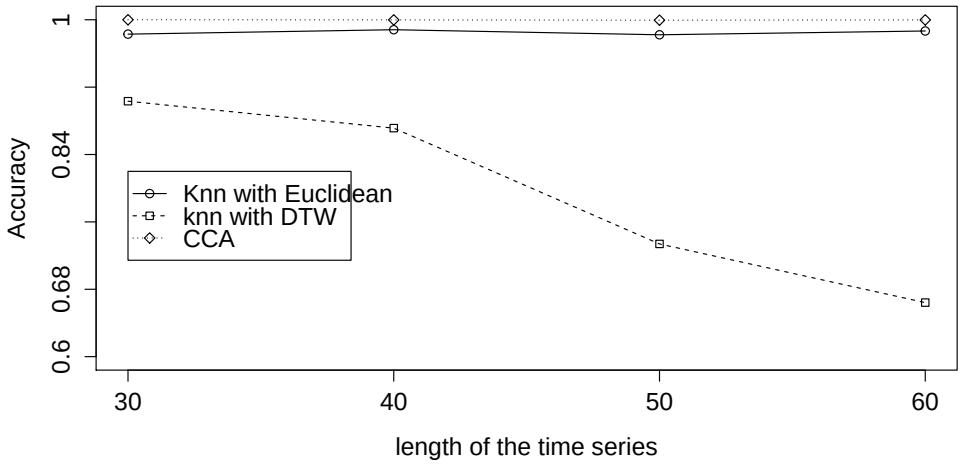

Figure 7.3. Accuracy for each of the algorithms, depending on the number of dimensions i.e. the length of the time series, with $k = 1$ for k-NN Intersection Algorithm

Accuracy shows the overall correctness of an algorithm. I can see that the accuracy drops drastically with the increase of the length of time series for CCA. Whereas, the accuracy of Knn Intersection algorithm for both the distance metrics are almost constant and always outperforms the accuracy of CCA.

7.4.2. Precision

Precision of an algorithm is defined as,

$Precision = \frac{NumberofTruePositives}{NumberofTruePositives+NumberofFalsePositives}$

precision is the ratio of the number of cases where time series pair were correctly detected to be related to the number of cases where our algorithm claimed that the pair of time series were related.

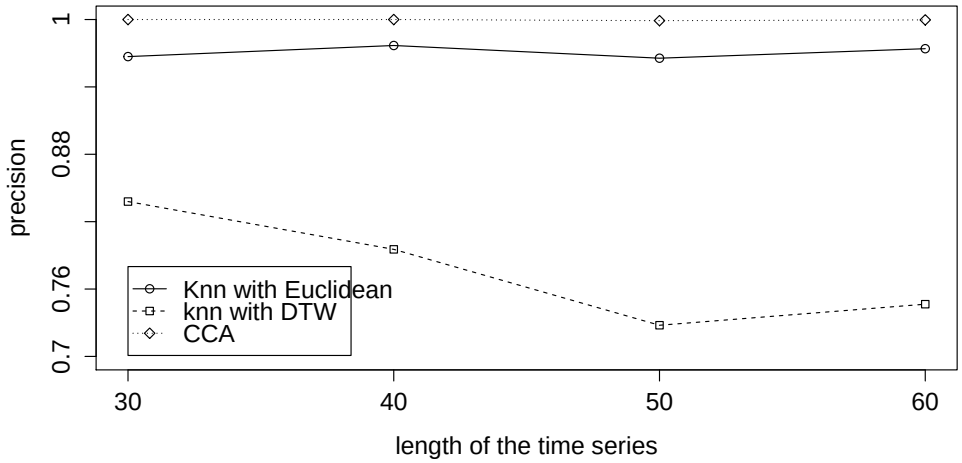

Figure 7.4. Precision for each of the algorithms, depending on the number of dimensions i.e. the length of the time series, with $k = 1$ for k-NN Intersection Algorithm

The above figure shows that both the Knn algorithms show higher Precision compared to that of Canonical Correlation Analysis. The Precision for Knn Intersection with Euclidean distance drops a little with the increase in the length of the time series. Whereas, Knn Intersection with dynamic time warping is very robust.

7.4.3. F1 Score

F1 score is the geometric mean of the Precision and Recall. $F1 = \frac{2*Precision*Recall}{Precision+Recall}$ Where, $Recall = \frac{NumberofTruePositives}{NumberofTruePositives+NumberofFalseNegatives}$

Both Precision and Recall are important metric to measure the correctness of the algorithm. F1 score is sometimes preferred over both Precision and Recall as it considers the trade off between Precision and Recall.

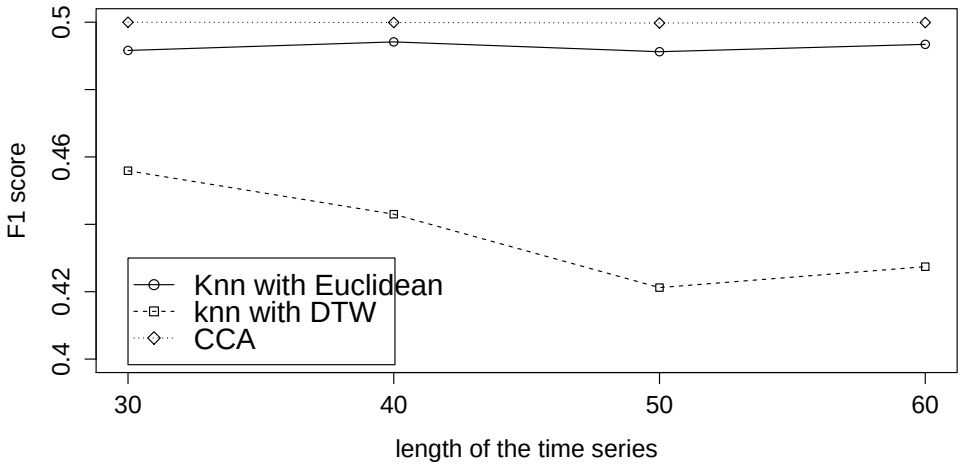

Figure 7.5. F1 score for each of the algorithms, depending on the number of dimensions i.e. the length of the time series, with $k = 1$ for k-NN Intersection Algorithm

The above figure shows that with increasing number of dimensions, the performance of Canonical Correlation decreases; whereas the performance of k-NN Intersection Algorithm for both Euclidean and dynamic time warping distance remain consistently good. The performance of k-NN Intersection Algorithm for dynamic time warping distance is always slightly better than the same for Euclidean distance.

7.5. Dependence on Sample Size

I want to test the behavior of our algorithm for different sample sizes. For this purpose I have first repeated our algorithms for different sample sizes and different number of dimensions. Now for every unique value of sample sizes, I have averaged the performance metrics over all the different values of time series length. That makes the result unbiased to any specific size of time series length. For example, I know from the previous section that the kNN performs much better when sample size is considered larger. So, if I restrict the test of dependence on sample size to a large value of time series length then the result would be biased to longer time series.

Now the comparative results are reported test the effect of the sample sizes on the performance of the algorithm.

7.5.1. Accuracy

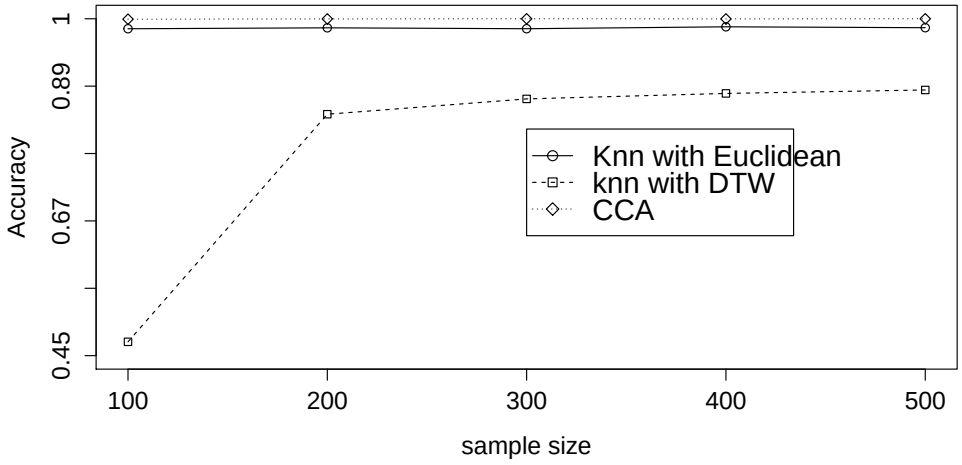

Figure 7.6. Accuracy for each of the algorithms, depending on the sample size i.e. the number of time series, with $k = 1$ for k-NN Intersection Algorithm

The above figure shows that Canonical starts to show a higher accuracy as the number of samples increases, but is still outperformed by the KNN Intersection. Dynamic time warping shows slightly better accuracy compared to Euclidean distance when it comes to KNN Intersection.

As discussed before, there as many applications where the time series data sets are High Dimensional Low Sample Size; CCA performs performs really bad for such applications.

7.5.2. F1 Score

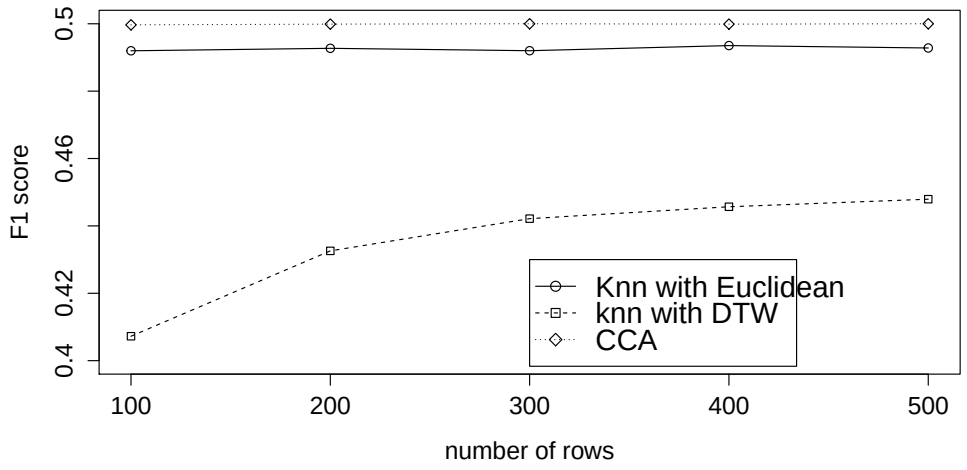

Figure 7.7. F1 score for each of the algorithms, depending on the sample size i.e. the number of time series, with $k = 1$ for k-NN Intersection Algorithm

The above figure shows that k-NN Intersection Algorithm always outperforms Canonical Correlation; but the gap in the performance is really high when the sample size is smaller. As and when the sample size increases, Canonical Correlation starts to perform better but is still outperformed by k-NN Intersection Algorithm. Similar to the plots for dependence on time series length, k-NN Intersection Algorithm for dynamic time warping distance always performs slightly better than the same for Euclidean distance.

k-NN Intersection Algorithm clearly out performs Canonical Correlation with relatively small number of rows, while it still performs pretty well even with large number of rows.

7.6. Runtime of the Algorithm

To perform a test on the runtime of our algorithm I have made a comparative study on both KNN Intersection algorithm and Canonical Correlation Analysis. I have considered only Euclidean distance for KNN Intersection algorithm because calculation of dynamic time warping distance between time series takes long time and our it should not be considered for speed issue.

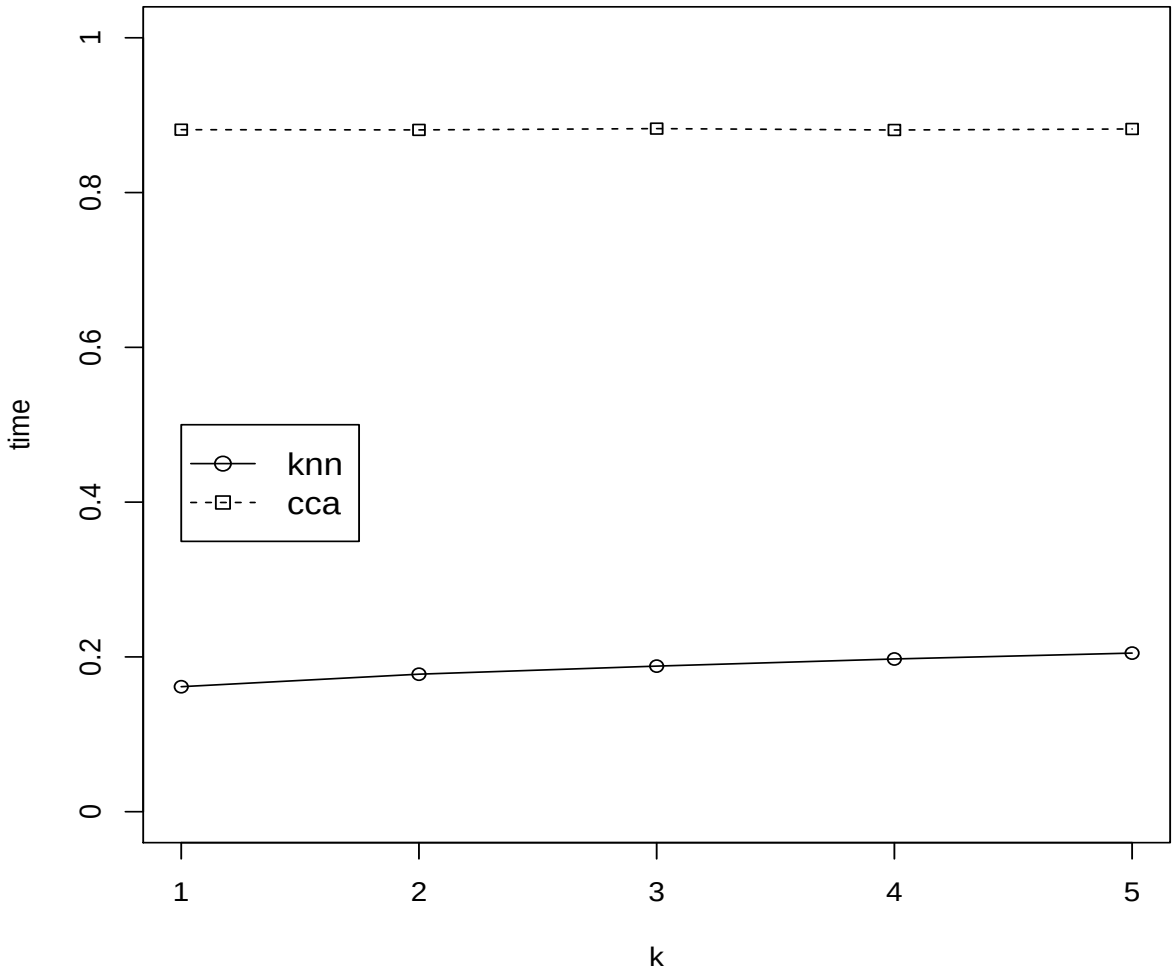

Figure 7.8. Average runtime depending on the model parameter k

I can see that the time consumed by KNN Intersection algorithm doesn't depend on the value of k.

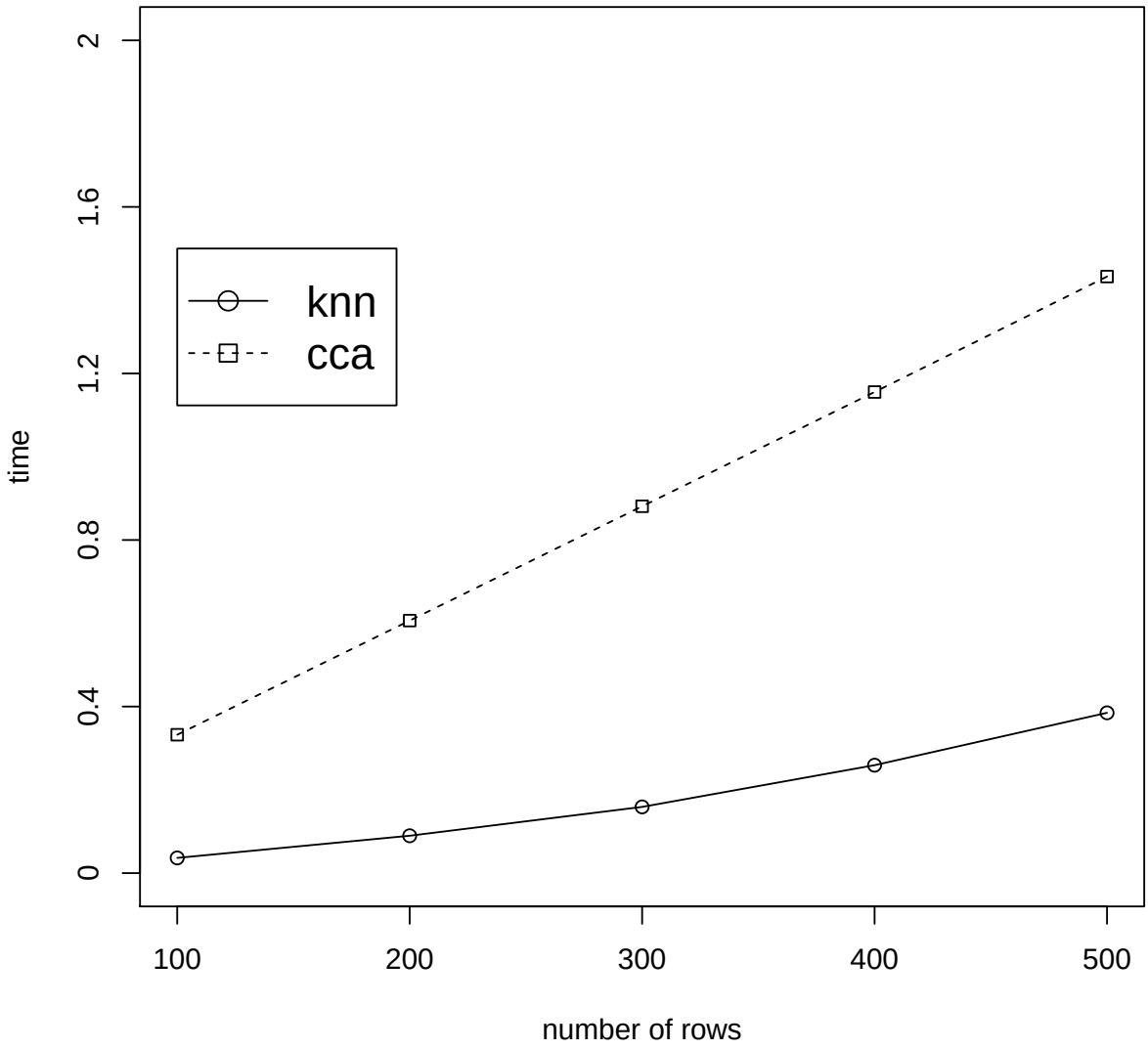

Figure 7.9. Average runtime depending on the sample size

Our algorithm clearly performs faster than Canonical Correlation even with smaller sample size. Canonical Correlation consumes linearly higher time with increasing number of samples. Whereas the time consumption increases very slowly with number of samples for our algorithm. There is big gap in time consumed for our algorithm with larger sample size which shows its efficiency for real world applications.

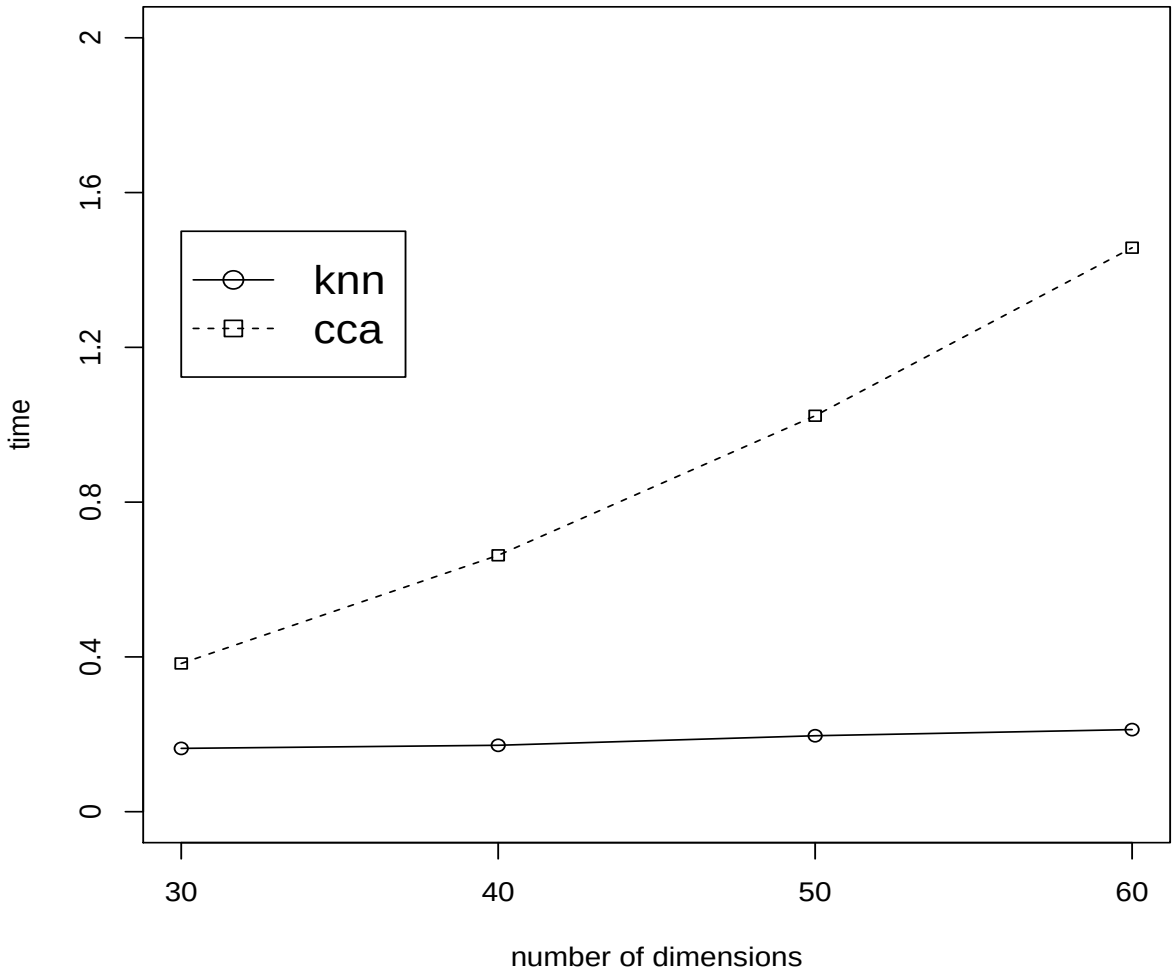

Figure 7.10. Average runtime depending on the length of the time series

With increasing number of dimensions, the time consumed for Canonical Correlation increases exponentially. Whereas our algorithm consumes almost constant time for increasing number of dimensions. That promises a clear edge on efficiency when it comes to real world time series data analysis because most of the time series data are high dimensional.

8. AGRICULTURAL DATA

As one application of our algorithm, we consider the effect of seasonal precipitation on plant growth. Understanding the influences of weather, in particular precipitation on vegetation enables prediction of productivity changes under different weather scenarios. [18] The relationship between precipitation and vegetation is strong and predictable when viewed at the appropriate spatial scale. There may not always be a strong correlation between weather variables and biomass. In many cases, NDVI time series is not related to precipitation or temperature time series [12]. That might mislead the analysts while trying to find significant difference in vegetation between regions based on the biosphere-atmosphere interactions [11]. For example, some extremely wet areas receive rainfall amounts in excess of a minimum precipitation threshold above which vegetation is unresponsive [17]. Thus, it is always a good idea to check the correlation between two time series data sets before proceeding with further analysis. That can help the environmental scientists decide whether to take another climate attribute into consideration for analyzing the vegetation growth trend. For the above scenario, temperature could be used to predict the effect on vegetation.

8.1. Test on Agricultural Data

I have used agricultural data to test our algorithm on a real life application. Plant growth is measured as the Normalized Difference Vegetation Index(NDVI) which is calculated from the density of green on a patch of land in the images collected by Satellite Imaging Corporation. Another time series data that I have considered here is the precipitation data. The precipitation time series is aggregated over monthly basis.

In the previous experiments, we used UCR time series data sets. We split the data sets vertically to produce two likely to be related data sets to test our algorithm. In the context of agricultural data, I consider the precipitation data to be the first data set which is the first half after the vertical split in the previous experiments. I call it the left data set. Each instance of the left data set is represented by one precipitation time series, sequence of aggregated precipitation values for consecutive time points, of a geo-spatial location. On the other hand, NDVI data set is considered as the second half of the original data set. I call it the right data set. And each instance

of the right data set is represented by one NDVI time series, NDVI values for consecutive time points, of the same geo-spatial location.

There are 3000 geo-spatial locations for which we have collected both the precipitation and the NDVI time series. For each iteration of execution, I have taken randomly sampled n locations from the 3000 available ones.

If the likelihood(measured collectively for all the geo-spatial locations) of overlap between the nearest neighbors (instances) of precipitation data and the nearest neighbors of NDVI data for the same geo-spatial location is significantly higher than the expected value by chance then we conclude that there is a significant relationship between precipitation and NDVI data sets.

I have tested the algorithm for variable lengths of precipitation time series. Suppose, we consider monthly precipitation data. In that case, each instance of the precipitation data set is of length 12 (starting from Jan,2013 to Dec,2013). And each value of the time series represents the aggregated precipitation value for a specific month. The length of NDVI time series data, on the other hand, have always been the same. It was collected bi-monthly for the month of June, July and August of 2013. So each instance of NDVI data set is of length 6 and the values of that time series represent the NDVI values from 1st June to 31st Aug, collected with the interval of 15 days.

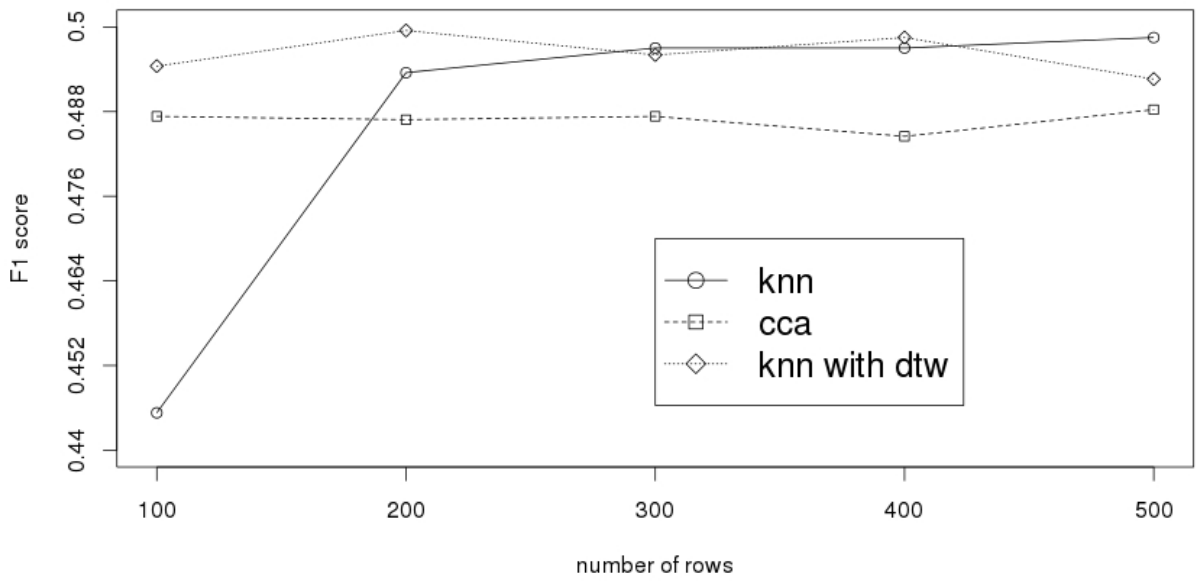

Figure 8.1. F1 score of each of the algorithms for agricultural data, depending on the sample size

I can see that the performance of KNN Intersection algorithm when used with Euclidean distance is improved with relatively higher sample size. Even for moderately high sample size, it performs better than Canonical Correlation Analysis. When dynamic time warping is used as the distance measure for KNN Intersection algorithm, it always gives exceptionally high performance.

8.2. Dependency on Sample Size

I intend to find the effect of sample size on KNN Intersection algorithm for agricultural data. I have only considered Euclidean distance here. As the input variable, I have considered the daily precipitation data and the NDVI index data as the output variable.

The error rate starts to drop with increasing sample size and flattens out from 300 sample points.

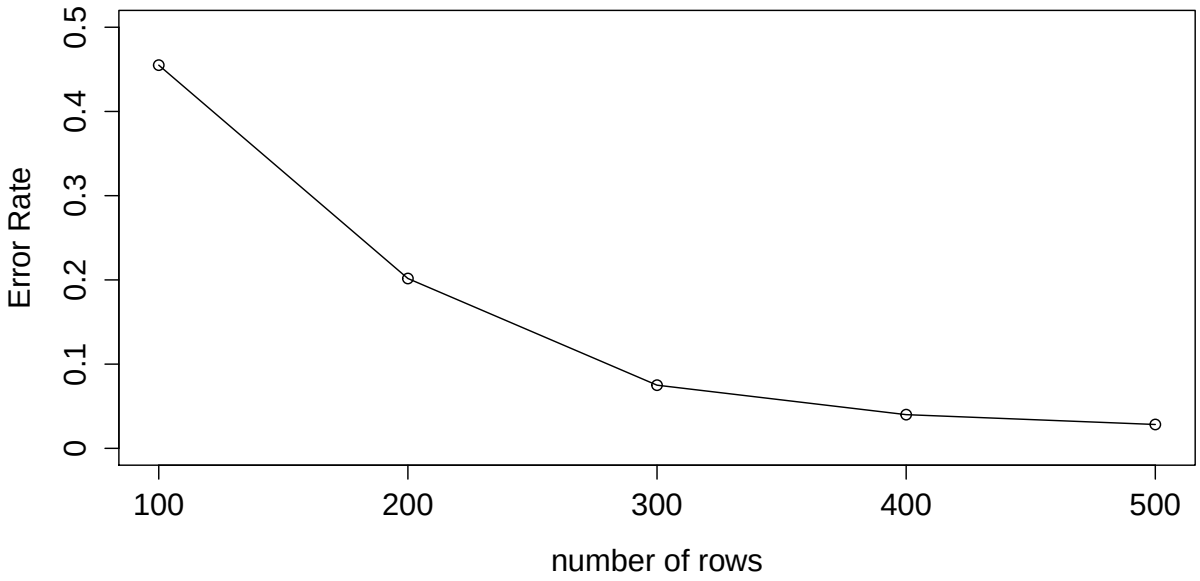

Figure 8.2. Error rate for agricultural data, depending on the sample size

8.3. Dependency on Time Series Length

High dimensional input data often leads to overfitting for machine learning algorithm. Aggregation of the time series data can help us reduce the dimensions without losing the data. Of course I will lose some pattern due to the aggregation. I intend to test the effect of aggregation for our algorithm. I have aggregated the daily precipitation data as weekly, biweekly and so on. For example, weekly precipitation data will lead to 50 dimensions of the input data. There is always an overlap of the time points between precipitation and NDVI time series data sets, though the ranges of the precipitation data is larger than that of NDVI data. I have used moderately high sample sizes(300,400,500) and averaged our result for all the considered sample sizes.

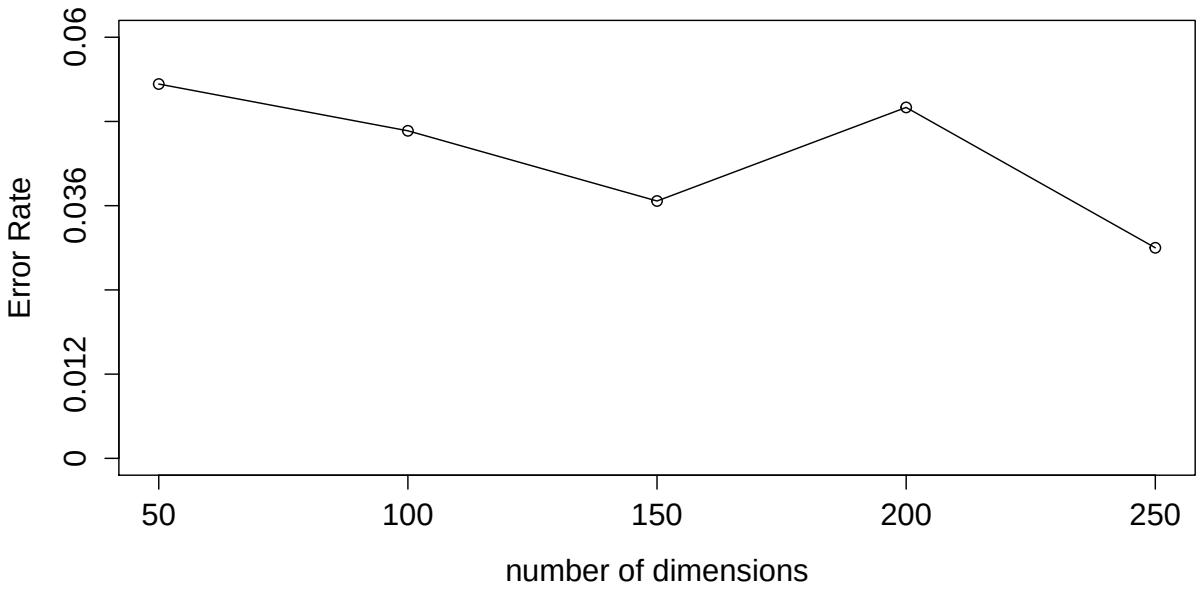

Figure 8.3. Error rate for agricultural data, depending on the sample size

The above figure illustrates that the increasing number of dimensions do not affect the performance of the algorithm (as the change in the error rate is negligible). So if I do not want to lose pattern by aggregating the time series data then I can always consider higher dimensions at the cost of computation time without letting oferfitting happen.

9. CONCLUSIONS

In summary, I have introduced a novel k-NN Intersection Algorithm and compared that with the existing Canonical Correlation Analysis for evaluating how strongly two time series data sets are related. I have shown that k-NN Intersection Algorithm showed better performance in both accuracy and computation time. I have further triggered dynamic time warping in the k-NN Intersection Algorithm and showed that it improves its accuracy throughout at the cost of computation time. Overall I have demonstrated the need for testing the strength of the relationship between two time series data sets, and I have presented an effective solution to the problem.

REFERENCES

[1] Shotaro Akaho. A kernel method for canonical correlation analysis. *CoRR*, abs/cs/0609071, 2006.

[2] Hirotugu Akaike. *[Mathematics in Science and Engineering] System Identification Advances and Case Studies Volume 126 —— Canonical Correlation Analysis of Time Series and the Use of an Information Criterion*. 1976.

[3] Hideki Asoh and Osamu Takechi. *An Approximation of Nonlinear Canonical Correlation Analysis by Multilayer Perceptrons*, pages 713–716. Springer London, London, 1994.

[4] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[5] Aaron French, Sally Chess, and Steve Santos. Canonical correlation & principal components analysis. *Online@ SFSU San Francisco State University http://userwww. sfsu. edu/˜ efc/-classes/biol710/pca/ccandpca. htm*, 2002.

[6] Peter Hall, J. S. Marron, and Amnon Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444, 2005.

[7] M. Han, R. Wei, and Decai Li. Multivariate chaotic time series analysis and prediction using improved nonlinear canonical correlation analysis. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 758–764, June 2008.

[8] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[9] William W. Hsieh. Nonlinear multivariate and time series analysis by neural network methods. *Reviews of Geophysics*, 42(1):n/a–n/a, 2004. RG1003.

[10] W.W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 13(10):1095 – 1105, 2000.

[11] Zhang Jingyong, Dong Wenjie, Fu Congbin, and Wu Lingyun. The influence of vegetation cover on summer precipitation in china: A statistical analysis of ndvi and climate data. *Advances in Atmospheric Sciences*, 20(6):1002, 2003.

[12] A. Kawabata, K. Ichii, and Y. Yamaguchi. Global monitoring of interannual changes in vegetation activities using ndvi and its relationships to temperature and precipitation. *International Journal of Remote Sensing*, 22(7):1377–1382, 2001.

[13] P. L. LAI and C. FYFE. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000. PMID: 11195936.

[14] J. Gary Lutz and Tanya L. Eckert. The relationship between canonical correlation analysis and multivariate multiple regression. *Educational and Psychological Measurement*, 54(3):666–675, 1994.

[15] Alvin C Rencher. *Methods of multivariate analysis*, volume 492. John Wiley & Sons, 2003.

[16] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[17] P.A. Schultz and M.S. Halpert. Global correlation of temperature, ndvi and precipitation. *Advances in Space Research*, 13(5):277 – 280, 1993.

[18] J. Wang, P. M. Rich, and K. P. Price. Temporal responses of ndvi to precipitation and temperature in the central great plains, usa. *International Journal of Remote Sensing*, 24(11):2345–2364, 2003.

[19] W. Wang, R. Arora, K. Livescu, and N. Srebro. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 688–695, Sept 2015.

[20] Alexander Zimmermann, Beate Zimmermann, and Helmut Elsenbeer. Rainfall redistribution in a tropical forest: Spatial and temporal patterns. *Water Resources Research*, 45(11):n/a–n/a, 2009. W11413.