A New Method for Pattern Classification through Dimensionality Reduction Based on Regression Analysis

Lalitha Rangarajan & P. Nagabhushan
Department of Computer Science, University of Mysore
lalitha.arun@mailcity.com & pnagabhushana@hotmail.com

This research is supported by ISRO project under RESPOND (10/4/317)

Abstract

A new pattern recognition methodology for classification of multidimensional, temporal samples is proposed. While a number of features (dimensions) of the image are required for better description of the image, they pose a threat in terms of unwieldy mass of data, particularly in the case of temporal observations. In this paper we have proposed a method to achieve dimensionality reduction of multi spectral temporal image using regression. In temporal images, each feature of a sample is not just a single numerical value, but a vector of real values. The method proposed finds the pattern of change in the feature values over time by fitting regression curves. A clustering methodology based on a new distance measure between fitted regression curves, is also proposed. The method suggested appears very versatile as it is readily applicable to any temporal image, be it gray scale, color or multi spectral. The algorithm is tested successfully on 2 different data sets.

Key Words: Pattern Classification, Dimensionality Reduction, Regression, Clustering,
Data Assimilation.

1. Introduction

Temporal imaging/observations uses time as an aid in image classification. An efficient method for dimensionality reduction of multi spectral temporal image is suggested here. Measurements made in pattern recognition applications are inherently multi dimensional in nature. Larger the number of features, more severe are the problems of storage and analysis time requirements [2]. Hence much importance has been attributed to the process of dimensionality reduction or feature reduction. Aim of dimensionality reduction is to describe the samples by means of a minimum number of features that are necessary for discrimination of objects in the image. A variety of methods for dimensionality reduction are proposed in literature [2,3,6]. Most of these belong to either subsetting methods [1], feature space transformation methods [3,6], or principal component analysis [2]. A survey of many dimensionality reduction procedures can be found in [8]. Unlike one time image where a feature of a sample is just a numerical value, in temporal images every feature of a sample is a vector of real values because the same image is observed at different time steps. Some of the methods to study temporal images use thresh holds on feature values for pattern classification [4], obtain classification maps of successive time steps and use them all for a classification of total data set [5].

We have made an attempt to summarize these huge sets of each feature values of a sample in shorter sets of new “units” namely regression curves. This results in data assimilation of temporal values of feature values of a sample.

Section 2 outlines the method developed. Details of finding regression lines are described in section 3. In section 4 we have discussed the computation of ‘distance’ between transformed samples which is the foundation of clustering the samples. Section 5 describes the clustering methodology. Section 6 outlines the experiments and the results obtained. In sections 7 and 8 are conclusions and further improvements possible to the method are suggested.

2. Overview of the method

One of the effective means of classifying temporal data is multi temporal profile approach [1]. The use of change detection for classification is the purpose of temporal imaging. Samples which are seemingly alike initially may tend to vary to a large extent at a later time. Two samples can be called “similar” only if they are similar over time. Hence we have developed a method to study the behavior of samples over time and use this to do pattern classification. Our method detects the changes in feature values of the samples over time and uses these changes for classification. The change in each of the feature values of a sample is described by a regression curve. We have used “least squared” error regression curve for multi temporal data. The s sample items, each having f features, over t time steps (size of the data is s*(f.t)) can be reduced to s*f regression curves. In other words (f.t) features of the sample items can be reduced to f regression curves. The transformation results in the assimilation of (f.t) features to f featured regression curves. A good regression curve provides an apt description of the samples. But it is difficult to determine the nature of regression curves. Even if we do, the j th feature values of different samples could yield different types of regression curves. For instance the t values of j th feature of samples i and k may become quadratic and a cubic curve. This amounts to keeping track of regression curves and also the type of regression curves. This results in too much of book keeping, making the problem of classification too tedious. Probably this may even contradict the possibilities of the very theme of the research, that is, dimensionality reduction.

Such as an arbitrary curve can be approximated by piecewise linear segments [9,10], a regression curve is represented by piecewise linear regression line segments. The time axis of the i th sample, j th feature has been sliced and regression lines identified in each slice. The number and length of the slices are kept uniform for all features and samples, to ease the problem of measuring distance between samples. In short s*(f.t) data items have been reduced to s*(f .c) regression lines, where c is the number of slices of time axis. A new distance measure has been proposed to measure the distance between the set of regression lines.

3. Method in detail

Let the data items be represented by d[i,j,k] where i the sample, j is the feature, k is the time step and $1 \leq i \leq s$, $1 \leq j \leq f$, $1 \leq k \leq t$. i th sample, j th feature values {d[i,j,1], d[i,j,2], ...d[i,j,t]}is divided (cut) into c slices each slice having t/c data items. Regression lines are identified for data sets {d[i,j,1], d[i,j,2], ... d[i,j,t/c]}, {d[i,j,t/c+1], d[i,j,t/c+2], ... d[i,j,2t/c]}, ... {d[i,j,(c-1)t/c], ... d[i,j,t]}.Thus d[i,j,k] are transformed into c regression lines of the form 'd' = a + b't'. where 'd' is the dependent variable of feature values and 't' is the independent variable of time. The above process of splitting t time steps into c slices and finding regression lines in all slices is repeated for all features and samples. The algorithm can be best understood with the example, where for a sample the number of features are 2, time steps are 6 and number of time slices is 2, illustrated in the figures 1and 2 in the next page.

Algorithm 1: For representation of data using regression lines.

Input: d[i,j,k]: $1 \leq i \leq s$, $1 \leq j \leq f$, $1 \leq k \leq t$, and c the number of slices of time axis.
Output: The coefficients of regression lines, a[i,j,p] (y intercept), b[i,j,p] (slope) for
$1 \leq i \leq s$, $1 \leq j \leq f$, $1 \leq p \leq c$.

```
1. For all samples i = 1 to s
2. For all features j = 1 to f
3. For all slices p = 1 to c
4. Find regression lines for the data set T = {(p-1)t/c
   +1, (p-1)t/c +2, ... pt/c},   D = {d[i,j,(p-1)t/c +1],
   d[i,j,(p-1)t/c +2], ... d[i,j,pt/c]} to be 'd' = a[i,j,p]
   + b[i,j,p] * 't'
5. Return (a[i,j,p], b[i,j,p] )
   End 3
   End 2
   End 1
```

4. Computation of distance measure

Two samples m,n are similar when d[m,j,k] and d[n,j,k] are close for all features and time step. That is d[m,j,k] and d[n,j,k] are approximately equal for $1 \leq j \leq f$, $1 \leq k \leq t$. The distance between m and n has been illustrated with an example in figures 3 and 4 in the next page.

Let L[sam,fea,sl], R[sam,fea,sl] be points of intersections of the regression line of sample 'sam' for feature 'fea' in the time slice 'sl' with the ordinates at left (beginning) and right (end) of the slice 'sl'.That is L[sam,fea,sl] R[sam,fea,sl] is the regression line of sample 'sam' for the feature 'fea' in the time slice 'sl'. It is clear that m, n are similar whenever L 's of samples m,n and R 's of samples m,n are close on the respective ordinates. That is whenever the distance between L 's and the distance between R's of the samples, on the respective ordinates are small. Our distance measure is a simple dissimilarity measure that is taken to be the maximum of the following distances {L[m,1,1] L[n,1,1], R[m,1,1] R[n,1,1], L[m,1,2] L[n,1,2], R[m,1,2] R[n,1,2], ... ,L[m,1,c] L[n,1,c],R[m,1,c] R[n,1,c], ... L[m,f,1] L[n,f,1], R[m,f,1] R[n,f,1], L [m,f,2] L[n,f,2], R[m,f,2] R[n,f,2], ... , L[m,f,c] L[n,f,c], R[m,f,c] R[n,f,c]}. That is distance between samples m,n can be denoted by dis[m,n] = Max { L[m,j,p] L[n,j,p], R[m,j,p] R[n,j,p] where $1 \leq j \leq f$, $1 \leq p \leq c$}. A smaller value of the above distance implies there is not much of a deviation of the respective sets of regression lines and hence the samples themselves are less deviated through out and can be called similar. A bigger value of the above distance implies there is a feature and a time slice p where the regression lines are so different because those feature values are not close and thus they are dissimilar.

The sample with 2 features and 6 time steps.

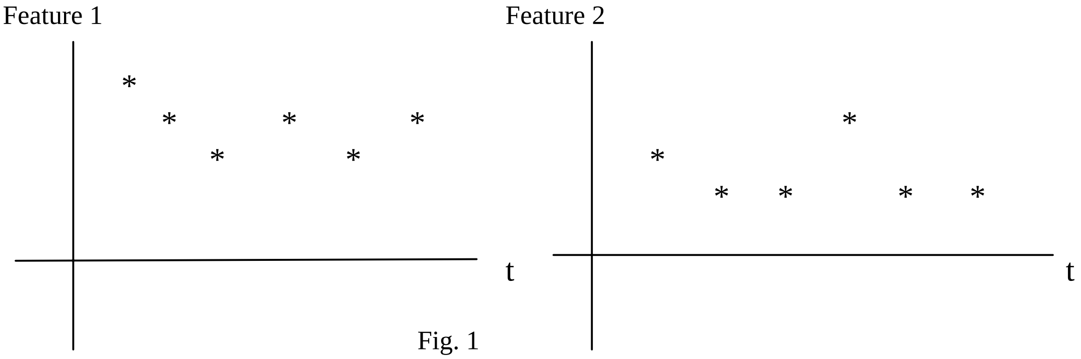

Fig. 1

Algorithm 1 may find following lines.

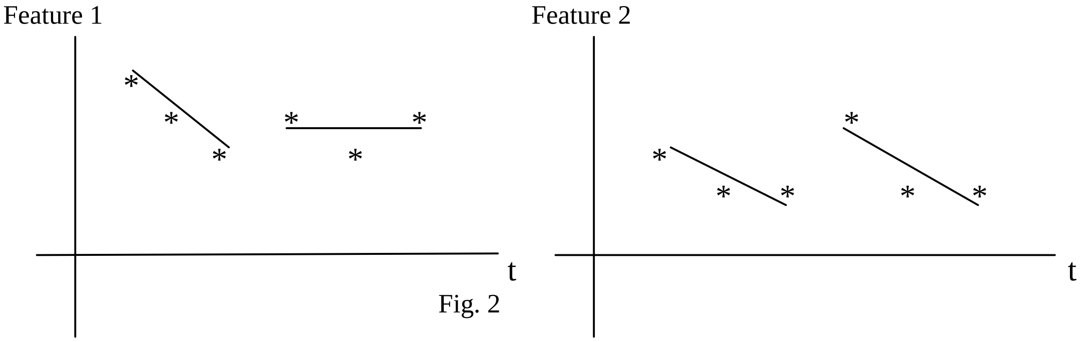

Fig. 2

Distance between 2 samples.

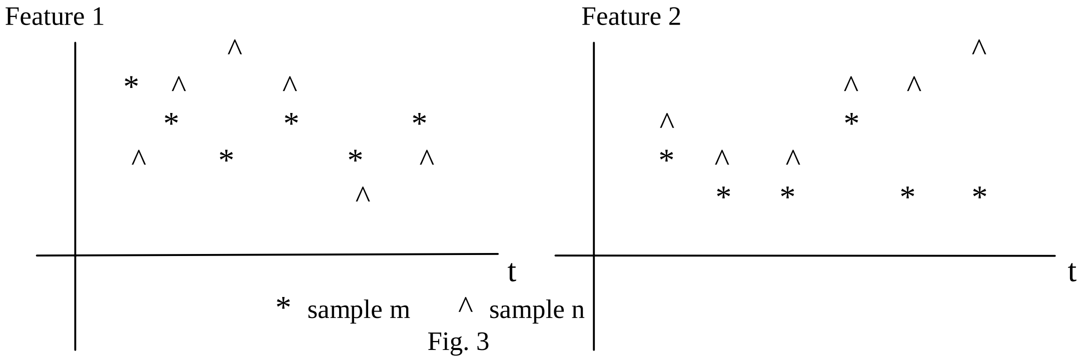

Fig. 3

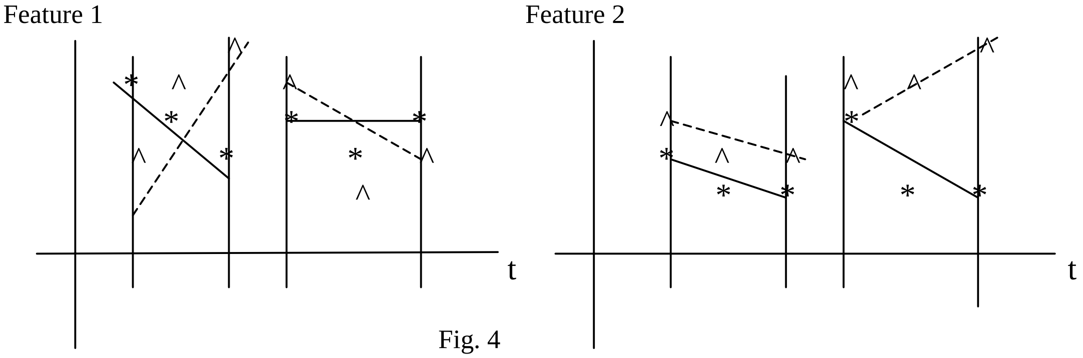

Fig. 4

Dis(m,n)=max{L[m,1,1]L[n,1,1],R[m,1,1]R[n,1,1],L[m,1,2]L[n,1,2],R[m,1,2]R[n,1,2], L[m,2,1]L[n,2,1],R[m,2,1]R[n,2,1],L[m,2,2]L[n,2,2],R[m,2,2]R{n,2,2]}.

L[m,fea=1,slice=1] / R[m,fea=1,slice=1] are the points of intersection of the regression line of m th sample for fea 1 and slice 1, with the ordinate at t=1 / 3 (left / right or beginning / end of the time slice).

5. Clustering

We have done a partitional clustering which is a combination of the methods by Forgy.E.W [3] and MacQueen.J.B [3]. The selection of initial seed points or initial cluster centres is done as suggested by Ball.G.H and Hall. D.J [3] in the following steps.

Algorithm 2: For finding seed points (seed lines) of the data set.
Input: Data in the form of regression lines.
Output: All seed points (centroids) of clusters (the number of clusters determined by the algorithm)

1. Take the overall mean of the data as first seed point.
2. Select subsequent seed points by examining the data units in their input sequence. Accept any data unit which is at least at some specified distance, DIS, from all previously selected seed points.
3. Perform 2 until no more seed points can be found.

For our data set, which is a set of regression lines, the initial seed point is to be selected as the line joining average of L[i,j,p], and the average of R[i,j,p] where $1 \le i \le s$, $1 \le j \le f$, $1 \le p \le c$. Mean is a set of lines joining the averages of L values and R values. In the simple example of figure 5 in the next page, where s=2, f=2, t=6, l=2, we have illustrated the computation of the centre lines. Suppose that the centre is denoted by the vector C[i,j] where i is the centre or the seed point number and j is the component number. For performing step 2 of Algorithm 2 we need distance between a centre and a sample (which is represented by a set of regression lines). In the figure 5 in the next page distance between centre C and sample 1 is defined as Max {L[1,1,1]C[1,1], R[1,1,1]C[1,2], L[1,1,2]C[1,3], R[1,1,2]C[1,4], L[1,2,1]C[1,5], R[1,2,1]C[1,6], L[1,2,2]C[1,7], R[1,2,2]C[1,8]}. In general distance between centre m and sample n can be stated as Max {L[n,j,p] C[m,(j-1)c + 2p-1], R[n,j,k]C[m,(j-1)c + 2p] where $1\le j \le f$, $1\le p \le c$}. After selection of initial seed points, initial partition is generated using Forgy's method. C be the centroid currently.LR be the regression lines of the samples for various time slices. New centroid NC for a particular time slice is the line joining, average of L's and C of that slice and the average of R's and C of that slice.

Algorithm 3: To find initial partition.
Input: Data in the form of regression lines, all seed points of clusters.
Output: Initial cluster of data sets.

1. Allocate each data unit to the cluster with the nearest seed point. The seed points remain fixed in this process.
2. Compute new seed points as centroids of clusters.

Step 2 of Algorithm 3 needs centroids of clusters which we computed as the vector whose components are average of all L and R values on all ordinates.
For obtaining refinement of the initial partition and final cluster MacQueen's k-means method is used.

Algorithm 4: To find final clusters.
Input: Initial cluster.
Output: Final cluster.

1. Start with initial partition described in the Algorithm 3 and the new centroids as seed points.
2. Determine the cluster of each data unit to be that with the nearest seed point. Re compute new seed points whenever data is added to or deleted from a cluster.
3. Perform 2 until no data changes it's cluster membership.

We have selected Forgy's method for initial partition, since a coarse partition will suffice to start with. The number of refinements is reduced comparatively with MacQueen's method of re computing the seed points whenever there is a change in cluster assignment.

6. Experiments

6.1. Temperature Data

Average of daily minimum and maximum temperatures of each month of 37 cities all over the globe is considered to be data with 2 features (minimum, maximum) and having 12 time steps. A small portion of the data is given in table 1 of the next page.

Experiments were performed with different numbers of time slices (2,3,4) and with various values of DIS (distance between seed points). In general all resulted in a satisfactory classification. In particular classification with 3 slices and DIS =10 (one tenth of maximum distance between samples) was very encouraging and placed cities which are hot and humid like Madras, Calcutta, Bombay, Kuala Lumpur, Hongkong, Singapore, Colombo etc in one cluster. Places which are very cold like Moscow, Munich, Stockholm, Toronto etc in a cluster. Cities where minimum temperatures are low and maximum is on the high side, like Frankfurt, Zurich are grouped together. Cities which have moderate

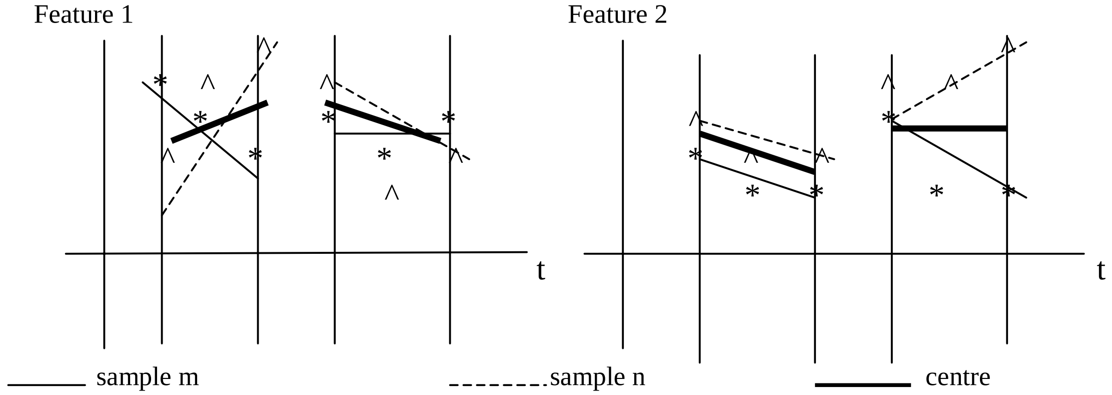

Fig. 5

Average Temperature data of Experiment 1(in Celsius)

CITY	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Amsterdam												
Min	-4	-5	2	5	7	10	10	12	10	5	1	-1
Max	4	3	12	15	17	20	20	23	20	15	10	4
…		…	…	…	…	…	…	…	…	…	…	…
Zurich												
Min	-11	-8	-7	-1	2	6	10	8	5	3	0	-11
Max	9	15	18	21	27	30	31	25	23	22	19	15

Table 1

A portion of the table of values of Experiment 2 are given below.

Date Feature Vegetation	Jun 11 1 - 6	Jun 29 1 - 6	Jul 16 1 - 6	Aug 30 1 - 6
Corn	3.7 -- 19.8	2.1 -- 6.4	2.2 -- 5.1	2.2 -- 4.3
Soybean	3.3 -- 16.9	2.2 -- 6.4	2.5 -- 10.2	2.2 -- 7.0

Table 2

temperatures in summer (not too high) and in winter (not too cold) belong to a cluster. The number of clusters is 6 in number. The general laws like increase in reliability of classes with a increase in number of time slices, and increase in number of classes with decrease in DIS, were observed.

6.2. Corn Soybean Data

This is also a temporal data collected to discriminate corn and soybean fields. The data was collected on unequal intervals of time. The time interval was suitably scaled. Incidentally our method of representation of data with regression lines also provide for interpolation or prediction of missing data. The data has 6 features and 4 time steps and a portion of this is given in table 2 of this page. The classes are previously known to be corn or soybean field. A single regression line for a feature (6 regression lines for a sample) resulted in a very good classification. The misclassification is 12% (4 out of 30 soybean fields is identified as corn field)

7. Conclusion

The problem of reducing the dimension of multi spectral temporal data has been addressed in a novel way. In the proposed method temporal values of each feature is represented as regression line/s. Thus feature values of samples are no more just ordinal values (difficult to store and understand) but transformed to new “units” namely “regression lines” (temporal data assimilated in the form of lines). Experiments with real data have demonstrated the ability of this method to produce natural classifications, which is the ultimate goal of multi spectral image processing. Potential applications of this method include areas such as remote sensing, medicine, agriculture, computer vision where multi temporal observations are encountered.

8. Further improvements

Instead of dividing the time step into fixed number of slices, a method to find the number of regression lines needed for a set of values could be devised. In other words a method that detects the pieces of approximately linear segments could be developed/used to transform feature values into regression lines. If this is done a natural classification is created among the samples as soon as regression lines are identified. The number of regression lines as well as the lengths of these lines should be the same for all features if 2 samples are to be called as “similar”. This creates a classification initially and could be fine tuned later.

References:

1. **T.**M.Lillesand, Ralph W.Kiefer, Remote Sensing and Image Interpretation, John Wiley & Sons, New York, 1992.
2. I.T.Jolliffee, Principal Component Analysis, Springer Verlag, New York, 1986.
3. M.R. Anderberg, Cluster Analysis for applications, Academic Press, New York, 1973.
4. K.C.Gowda, P. Nagabhushan, “Stereo Graphic Projection based dimensionality reduction of remotely sensed multidimensional data”, International Seminar on Image Processing, Nov.1989.
5. D.N.Martono, and D. Dirgayahu, “The application of Remote Sensing for monitoring forest conditions in north Sumatera”, Asian-Pacific Remote Sensing Journal, vol.5, no.1, July 1992.
6. H.N. Srikanta Prakash, P.Nagabhushan, and K.C. Gowda, “Symbolic Data Analysis of multi temporal data: An application to seasonal analysis”, IEEE International Geoscience and Remote Sensing symposium, Singapore,3-8 Aug 1997.
7. Kulkarni Linganagouda, P.Nagabhushan, and K.Chidananda Gowda, “A new approach for Feature transformation to Euclidean space useful in the analysis of Multispectral data”, IEEE IGRASS-94, 1994.
8. P. Nagabhushan, An efficient method for classifying remotely sensed data (incorporating dimensionality reduction), Ph.D thesis, University of Mysore, 1988.
9. Maylor K. Leung , Yee-Hong Yang,“Dynamic Strip Algorithm in Curve Fitting”,Computer Vision, Graphics, and Image Processing, vol 51, 1990.
10. Dunham J.G, "Piecewise linear approximation of planar curves", IEEE Trans. PAMI, vol 8, 1986.