Multiple Classifier Selection to Improve Accuracy of Classifier for Time Series Analysis

N.Yamuna Rani

Department of Computer Science and Engineering, Anna University, Coimbatore-641047

R.Velmani

Department of Computer science and Engineering, Anna University, Coimbatore-641047

ABSTRACT

In this article we proposed a technique for temporal data mining which is based on the classification rules and optimal discriminant analysis(ODA).Time series are decomposed into segments (avg,slope,curvature) are described by polynomial models. Then the classifier assesses subsequent segments based on the classification rule activity. And assign an input a class. Segmentation and piecewise polynomial modeling are done fast over time series. For this classifier we use Euclidean distance measure for time series and using a fast Fourier Transform (FFT) to construct a multiple dynamic classifier to increase the accuracy of a classifier.

Index Term

Fast Fourier transform, generative classifier, optimal discriminant analysis, piecewise polynomial representation, Temporal data mining, time series classification

1. Introduction

In this paper we suggest tasks of temporal data mining such as segmentation,classification,clustering,forecasting of time series.(See [1],[2],[3],[4],[5]) for more information on temporal data mining).

We describe an approach to discover and apply classification rules for time series, segments of time series. It will be shown how a classifier based on such classification rules [13].It considers the uncertainty associated with the occurrence of motifs by a gradual assessment of two segment's similarity.

Assume we have to classify a given segment with number l in a sequence of segments, for example. Then, one of the classification rules could look like

If slope (l) is high and curvature (l - 1) is medium
Then class $_1$ (l) is 0.90 and class $_2$ (l-1) is 0.10

This rule uses information about trends of the time series (here: slope and curvature) in two subsequent segments numbered l and l - 1 as inputs. The terms high and medium represent univariate Gaussian functions.Time series analysis based on the online segmentation algorithm [7]. Time series representation using piecewise polynomial of time series [14][15] .And a classifier based on Gaussian Mixture models [16][17]. I describe a similarity measure for time series or segments to build a Dynamic classifier from an already available static Gaussian Mixture Model.

The remainder of the article is structured as follows: Section 2 describes problem definition. Section 3 discuss about time series segmentation and generative models. Finally, Section 4, 5 summarizes the major findings and gives an outlook to future work.

2. Problem Definition

In our existing system analyzing the single property of a clustered object. It was trained by naïve Bayes classifier algorithm. It has major disadvantages

- Independence assumption wrong
  - Absurd estimates of class probabilities
  - Threshold must be tuned, not set analytically
- Generative model
  - Generally lower effectiveness than discriminative techniques (e.g. log. regress.)
  - Improving parameter estimates can *hurt* classification effectiveness

To overcome this problem we have choosen optimal discriminant analysis (ODA) algorithm. And to improve accuracy of classifier we have to consider all properties (or) features of a clustered object.

Manuscript received January 5, 2013
Manuscript revised January 20, 2013

3. Online Segmentation Algorithm

In this section, I have used an approach for time series segmentation using online segmentation algorithm and segmented time series representation using piecewise linear representation. **We will show**

1. How a time series can be segmented and how the segments can be modeled in a suitable form, **2.** How the similarity of time series or segments can be measured based on these models, and **3.** How rule sets (i.e., classifiers) of the form sketched in Section 1 can be found and applied to classify segments or sequences of segments.

3.1 Time Series Segmentation and Segment Representation

In [7] I described a technique for online segmentation algorithm for a very fast time series segmentation and modeling that yield a piecewise probabilistic representation—which in turn includes a piecewise polynomial representation—of a time series.

3.1.1 Piecewise linear representation of time series:

One of the most used representation of time series is piecewise linear approximation. It's frequently used in pattern recognition (similarity search, distance measures, clustering).

3.1.2 Linear interpolation and linear regression

In our work I have implemented the **SWAB** (Sliding Window and Bottom-up) algorithm and we have compared it with the Sliding Window and the Bottom-up algorithm.

Before going into details of the algorithm, we establish some preliminary notation:

- A time series *T* in the form *t1, t2, … , tn* is given
- *T[a:b]* denotes the subsection of *T* from *a* to *b*

Given that we approximate a time series with straight lines, there are at least two ways we can find the approximating line:

1. **Linear interpolation**: in this case the approximating line connects ta and tb, computable in constant time
2. **Linear regression**: in this case the approximating line best fits T[a:b] in the least square sense, computable linear in time according to the length of the segment

As we can see, interpolation shows a more aesthetical appealing because the endpoints are aligned (appropriate for computer graphics applications). Linear regression is maybe less appealing, but gives a tighter approximation (for example, when using the Euclidean Distance).

In the implementation of our algorithm we have chosen the linear interpolation, because of its aesthetic superiority and its low computational complexity. All segmentation algorithms require methods to evaluate the quality of fit. Frequently used similarity measures are:

- Sum of squares: taking all vertical differences between the best fit-line and the actual data points, squaring and summing up them
- Euclidean distance: square root of the sum of squares
- L∞ norm: distance between the best fit line and the data point furthest away in the vertical direction.

3.1.3 The sliding window algorithm: the straight forward approach

In this case a segment is grown until it exceeds some error bound [7]. The process repeats with the next data point not included in the newly created segment. In other words, starting from a point tk, include step by step a time point *t (k+1)*, *t (k+2)* extending the segment as long as the approximation error remains below a user specified threshold. If at the point *tj* the error exceeds the threshold, the subsequence *T [k: j-1]* is transformed into a segment and the procedure starts again at *tj*. It's an online algorithm, therefore often used in data mining tasks or in online patient monitoring, but it's not able to look ahead (there isn't a global view). The performance depends on type of data set; especially noise can over fragment the approximation. This algorithm is attractive because of its intuitiveness, simplicity and in particular for the fact that it is an online algorithm.

3.1.4 The top-down algorithm

Our time series is recursively partitioned until some stopping condition is met [7]. The time series is split at the best location and both subsequences are tested. If the approximation error is below a user-specified threshold this subsection is accepted; if the approximation error is above the threshold the subsection is splitted again until all the segments have approximation errors below the threshold.

The bottom-up algorithm this method is the complement to the Top-Down algorithm [7]. Starting from the finest possible approximation, segments are merged until some stopping condition is met. In other words, the entire time series *T* with n samples is split into *n/2* segments (each containing two time steps). Afterwards those segments are merged having the lowest merging costs until stopping criteria are met (for example, as merging cost it can be used the sum of squares).

3.1.5 Piecewise Polynomial Representation

Assume given a time series consisting of real valued observations (data points) y_n at points in time x_n with n€ {0, 1…N} we assume further that these data points are equidistant in time. Our model of this time series is a normal distribution with time-dependent mean and constant variance σ^2.

$$\mathbf{N(y|\varphi_w(x),\sigma^2)} \qquad \textbf{(1)}$$

Where x models the time and $\boldsymbol{\varphi_w}$ is a polynomial of degree K given by a linear combination (with weight vector w € IR^{k+1}) of certain basis polynomials $\boldsymbol{\varphi_k}$

$$\boldsymbol{\varphi_w(x)=\sum_{k=0}^{k} w_{k}\varphi_k(x)} \qquad \textbf{(2)}$$

With w_k€ IR, where the basis polynomials $\boldsymbol{\varphi_k(k€\{0,\dots,k\})}$ must have the following properties:
1. They must have different and ascending degrees 0; . . .; K.
2. The leading coefficient (coefficient of the monomial with the highest degree) of each basis polynomial must be one.
3. Each pair of basis polynomials $\varphi_{k1,}$ φ_{k2} (with k1≠k2) must be orthogonal with respect to the inner product

$$\boldsymbol{<\varphi_{k1,}\ \varphi_{k2}>=\sum_{n=0}^{N}\varphi}\mathrm{k1}\ (\boldsymbol{x1}), \qquad \textbf{(3)}$$

The so-called orthogonal expansion coefficients w_k are found by means of least-squares approximation techniques. The residuum of the approximation divided by the number of data points N + 1 gives the variance σ^2 of the probabilistic model.

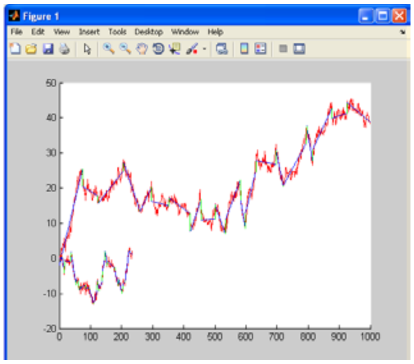

Fig. Example of a time series segmentation and their piecewise linear representation.

3.2 Euclidean distance Measure for two segments.

The **Euclidean distance** between point's **p** and **q** is the length of the line segment connecting them ($\overline{\mathbf{pq}}$)

In Cartesian product, if p = ($p_1, p_2,..., p_n$) and q = ($q_1, q_2,..., q_n$) are two points in Euclidean distance, then the distance

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2} = \backslash \qquad \textbf{(4)}$$

from p to q, or from q to p is given by:

The position of a point in a Euclidean *n*-space is a Euclidean vector. So, **p** and **q** are Euclidean vectors, starting from the origin of the space, and their tips indicate two points. The **Euclidean length** or **magnitude** of a vector measures the length of the vector:

$$\|\mathbf{p}\| = \sqrt{p_1^2 + p_2^2 + \cdots + p_n^2} = \sqrt{\mathbf{p}\cdot\mathbf{p}} \qquad \textbf{(5)}$$

A vector can be described as a directed line segment from the origin of the Euclidean space (vector tail), to a point in that space (vector tip). If we consider that its length is actually the distance from its tail to its tip, it becomes clear that the Euclidean norm of a vector is just a special case of Euclidean distance: the Euclidean distance between its tail and its tip.

$$\|\mathbf{q}-\mathbf{p}\| = \sqrt{\|\mathbf{p}\|^2 + \|\mathbf{q}\|^2 - 2\mathbf{p}\cdot\mathbf{q}}. \qquad \textbf{(6)}$$

3.3 Static classifier for Single segments.

Assume that we are now given a set of segments with samples v_l(l € {0,..,L}and L <<N) and corresponding class labels (targets) tl. These class labels describe the assignment of the segments to a class c € {1… C}. the vl are points in an I-dimensional feature space spanned by the Various attributes used for classification (orthogonal expansion coefficients, variances, and segment lengths, if needed).The set of labeled samples is now used as a training set to build a classifier for single segments. Here, the classification problem will be solved by means of a generative classifier based on a Gaussian mixture model (cf. [17], [18], for instance). The output of this classifier shall be interpretable as p(c|v), i.e., the distribution of classes given an input v. The components of the mixture model are denoted by j € {1,…,J}. Then, p(c|v) can be decomposed as follows:

$$\mathrm{p(c|v)} = \frac{\mathrm{p(v|c)p(c)}}{p(v)} \qquad \textbf{(7)}$$

$$= \frac{(\sum_{j=1}^{J} p(v|j)p(j|c))\mathrm{p(c)})}{\sum_{j=1}^{J} p(v|j')p(j')} \qquad \textbf{(8)}$$

$$\sum_{j=1}^{J} \frac{p(j|c)p(c)}{p(j)} . \frac{p(v|j)p(j)}{\sum_{j'=1}^{J} p(v|j')p(j')} \qquad (9)$$

- ✓ the p(j) are multinomial distributions
- ✓ for a given sample v', the p(j|v') are called responsibilities (i.e., of the component for the "Generation" of this sample), and
- ✓ .the p(c|j) can be seen as J multinomial distributions.

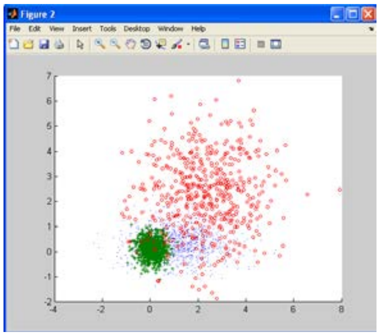

Fig. Example for a Gaussian mixture model.

3.4 Dynamic classifier for sequence of segments.

It will show how to expand the sequence of segments for dynamic classifier Using Fast Fourier Transform (FFT).

The FFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \dots, N-1. \qquad \textbf{(10)}$$

Evaluating this definition directly requires $O(N^2)$ operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in O(N log N) operations. More precisely, all known FFT algorithms require O(N log N) operations (technically, O only denotes an upper bound), although there is no known proof that better complexity is impossible.

To illustrate the savings of an FFT, consider the count of complex multiplications and additions. Evaluating the DFT's sums directly involves N^2 complex multiplications and $N(N-1)$ complex additions [of which $O(N)$ operations can be saved by eliminating trivial operations such as multiplications by 1].

Finally build a dynamic classifier by applying the technique called optimal discriminant analysis (ODA [9]) to the expanded Gaussian mixture model using the expanded samples.

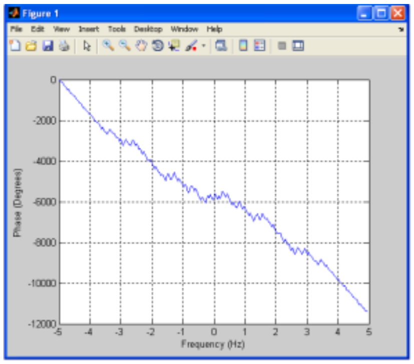

Fig. Example for Fast Fourier Transform (FFT)

3.5 Rule Extraction and Relation to fuzzy classifiers

The extraction of rules of the form shown in Table 1 is now straight forward. To see this, we have to consider that—due to the fact that we have diagonal covariance matrices—each multivariate exponential function contained in a Gaussian component can also be written as a product of univariate exponential functions. These univariate Gaussians weighted with the mixing coefficients and normalized, correspond to the functions' of the rules; the remaining scalar factors (the parameters of the component-conditional class distributions) correspond to The scalars w. It should also be emphasized that the specific way to construct the dynamic classifier from a static one keeps the number of different univariate Gaussians low (cf.The example below). Altogether, the dynamic classifier aims at containing rules that are comprehensible for human domain experts.

Three rules:
If average (l) is low and average (l - 1) is high
Then lp is 1:0 and hplf is 0:0 and hprf is 0:0
If average (l) is high and average (l -1) is high
Then lp is 0:0 and hplf is 0:0 and hprf is 1:0
If average(l) is high and average(l - 1) is low
Then lp is 0:0 and hplf is 1:0 and hprf is 0:0

The terms “low” and “high” are modeled by the corresponding univariate Gaussian functions. It should be emphasized that the required number of three rules is found by the ODA[9] technique itself. Due to the specific technique of constructing the rule premises for the dynamic classifier, we only need a number of two different univariate Gaussians for each input dimension, average(l) and average(l - 1).

At first glance, this kind of rule sets seems to describe a fuzzy classifier (cf, e.g., [22]). Actually, we have shown in [16], [23] that a static classifier as the one described above is functionally equivalent to fuzzy classifiers under some mild assumptions, e.g., certain Takagi-Sugeno fuzzy classifiers with Gaussian membership functions. The same holds for the dynamic classifier, of course. Thus, both would basically be able to achieve the same classification results. The meaning of the two kinds of classifiers, however, is very different: Fuzzy systems are based on the theory of fuzzy sets whose elements have membership degrees. Fuzzy sets permit a gradual assessment of the membership of elements of a set by means of so-called membership functions valued in the unit interval. Our dynamic classifier is based on probability theory. A gradual assessment of inputs of the classifier is effected by means of a mixture density model.

With both kinds of classifiers it is possible to deal with uncertainty concerning the occurrence of a certain motif.

4. Conclusion and Outlook

In this paper, we presented an approach to temporal data mining. The approach combines a technique for time series segmentation and representation with a method for building a dynamic, generative classifier.

4.1 Online segmentation algorithm advantages

It has three advantages. First advantage is to be able to detect frequent time-series patterns fast because it don’t need to keep whole time-series data. Second advantage is to be able to report the number of categories of time-series patterns. Third advantage is to acquire new category patterns incrementally.

The former usually produces much fewer segments and is faster and more reliable in the running time than other methods. The latter can reduce the representation error with fewer segments. It achieves the best overall performance on the segmentation results compared with other methods.

4.2 Optimal Discriminant Analysis (ODA) advantages

Discriminant Analysis may be used for two objectives: either we want to assess the adequacy of classification, given the group memberships of the objects under study; or we wish to assign objects to one of a number of (known) groups of objects. Discriminant Analysis may thus have a descriptive or a predictive objective.

In both cases, some group assignments must be known before carrying out the Discriminant Analysis. Such group assignments, or labeling, may be arrived at in any way. Hence Discriminant Analysis can be employed as a useful complement to Cluster Analysis (in order to judge the results of the latter) or Principal Components Analysis.

4.3 Advantages over this article:

To improve accuracy of a classifier we analyzing all feature space of the object .It should be trained using the technique called Optimal Discriminant Analysis is rather easy—the classifier is robust in the sense that a wide range of values can successfully be applied.

5. Future Work

5.1 Future work replaces the generative classifier by an averaged one-dependence estimator (AODE)

A probabilistic classification learning technique. It also developed to address the attribute-independence problem of the popular naive Bayes classifier. It also frequently develops substantially more accurate classifiers than naive Bayes at the cost of a modest increase in the amount of computation.

5.2 Advantage of An averaged one-dependence estimator (AODE) over optimal discriminant analysis (ODA)

However, because the base probability estimates are each conditioned by two variables rather than one, they are formed from less data (the training examples that satisfy both variables) and hence are likely to have more variance. AODE reduces this variance by averaging the estimates of all such ODEs.

References

[1] C. Antunes and A. Oliveira, “Temporal Data Mining: An Overview,” Proc. Workshop Temporal Data Mining, Knowledge Discovery and Data Mining (KDD ’01), pp. 1-13, 2001.

[2] J.F. Roddick and M. Spiliopoulou, “A Survey of Temporal Knowledge Discovery Paradigms and Methods,” IEEE Trans.Knowledge and Data Eng., vol. 14, no. 4, pp. 750-767, July/Aug.2002.

[3] S. Laxman and P.S. Sastry, “A Survey of Temporal Data Mining,”Sadhana, vol. 31, no. 2, pp. 173-198, 2006.

[4] Q. Zhao and S. Bhowmick, “Sequential Pattern Mining: A Survey,” technical report, anyang Technichal Univ., 2003.

[5] A.R. Post and J.H. Harrison, “Temporal Data Mining,” Clinics in Laboratory Medicine, vol. 28, no. 1, pp. 83-100, 2008.

[6] W. Hsu, M.L. Lee, and J. Wang, Temporal and Spatio-Temporal Data Mining. IGI Publishing, 2007.

[7] Eamonn Keogh Selina Chu David Hart Michael Pazzani,” An Online Algorithm for Segmenting Time Series”.

[8] P. Maji and S. Pal, “Rough-Fuzzy C-Medoids Algorithm and Selection of Bio-Basis for Amino Acid Sequence Analysis,” IEEE Trans. Knowledge and Data Eng., vol. 19, no. 6, pp. 859-872, June 2007.

[9] Deng Cai, Student Member, IEEE, Xiaofei He, and Jiawei Han, Senior Member, IEEE,” An Efficient Algorithm for Large-Scale Discriminant Analysis”.

[10] F. Fassetti, G. Greco, and G. Terracina, “Mining Loosely Structured Motifs from Biological Data,” IEEE Trans. Knowledge and Data Eng., vol. 20, no. 11, pp. 1472-1489, Nov. 2008.

[11] Y. Kocyigit, A. Alkan, and H. Erol, “Classification of EEG Recordings by Using Fast Independent Component Analysis and Artificial Neural Network,” J. Medical Systems, vol. 32, no. 1,pp. 17-20, 2008.

[12] G. Incerti, E. Feoli, L. Salvati, A. Brunetti, and A. Giovacchini,“Analysis of Bioclimatic Time Series and Their Neural Network-Based Classification to Characterise Drought Risk Patterns in South Italy,” Int’l J. Biometeorology, vol. 51, no. 4, pp. 253-263, 2007.

[13] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan,“Detecting Time Series Motifs under Uniform Scaling,” Proc.13th Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 844-853, 2007.

[14] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick, “Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations,”IEEE Trans. Pattern Analysis and Machine Intelligence,vol. 32, no. 12, pp. 2232-2245, Dec. 2010.

[15] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick, “On-Line Motif Detection in Time Series with SwiftMotif,” Pattern Recognition,vol. 42, no. 11, pp. 3015-3031, 2009.

[16] D. Fisch, B. Ku¨ hbeck, B. Sick, and S. Ovaska, “So Near and Yet So Far: New Insight into Properties of Some Well-Known Classifier Paradigms,” Information Sciences, vol. 180, no. 18, pp. 3381-3401,2010.

[17] C.-H. Lee, A. Liu, and W.-S. Chen, “Pattern Discovery of Fuzzy Time Series for Financial Prediction,” IEEE Trans. Knowledge and Data Eng., vol. 18, no. 5, pp. 613-625, May 2006.

[18] C.M. Bishop, Pattern Recognition and Machine Learning. Springer,2006.

[19] M.W. Kadous, “Learning Comprehensible Descriptions of Multivariate Time Series,” Proc. 16th Int’l Conf. Machine Learning (ICML),pp. 454-463, 1999.

[20] P. Geurts, “Pattern Extraction for Time Series Classification,” Proc.Fifth European Conf. Principles of Data Mining and Knowledge Discovery (PKDD), pp. 115-127, 2001.

[21] E. Keogh and M. Pazzani, “An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback,” Proc. Fourth Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 239-241, 1998.

[22] H. Zimmermann, Fuzzy Set Theory and Its Applications, fourth ed. Kluwer Academic Publishers, 2001.

[23] O. Buchtala and B. Sick, “Goodness of Fit: Measures for a Fuzzy Classifier,” Proc. First IEEE Symp. Foundations of Computational Intelligence (FOCI ’07), pp. 201-207, 2007.

Yamuna Rani currently am doing my M.E (Software Engg) in Anna University, Coimbatore. I did my B.Tech (IT) in Maharaja Engg College, Anna University, Chennai. And am working toward to do my PhD in CEG.