MOTION PREDICTION USING ADAPTIVE ASSOCIATION RULES

Ing. Lukáš STRYKA, Doctoral Degree Programme (1)
Dept. of Information Systems, FIT, BUT
E-mail: xstryk00@stud.fit.vutbr.cz

Supervised by Dr. Jaroslav Zendulka

ABSTRACT

This paper presents a method for object motion prediction in video. The method will work on XML data produced by M4 Project. The M4 project (Multi Modal Meeting Manager) is concerned with the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analyzed meetings. Output XML data describe motions of some objects in video, but this data is not complete. So we need some prediction model which provides us information about position and movement of the objects in video sequences which are not described in output XML document.

This paper is focused on a prediction method using adaptive association rules. These rules are generated during time series processing and they continuously adapt to current behavior of the time series representing the motion vector of an object.

1 INTRODUCTION

The M4 project (Multi Modal Meeting Manager) [5] is concerned with the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analyzed meetings. One of the project's work programmes is action/gesture recognition in video files. Output data of the recognition is stored in a XML file. However, the data is not complete and there are some sequences with no position and motion description of recognized object.

If we want to predict motion of the object in video, we use some prediction method. In this paper is presented prediction method based on adaptive association rules. Association rules are used in knowledge discovery process.

Knowledge discovery in databases deals with the process of extracting interesting knowledge from large amounts of data, usually stored in large databases or data warehouses. Knowledge can be represented in many different ways, such as clusters, decision trees, decision rules, etc.; in particular, association rules have proved to be an effective tool to discover interesting relations in massive amounts of data. There are some methods for extracting association rules from XML data [2], [3], so adaptive association rules method can be inspired by these methods to work over the XML data.

2 TIME SERIES PREDICTION

We want predict a motion of object in video. We can decompose motion vector into two separated components - horizontal and vertical vector. Than we can work with each component separately like with one-dimensional time series.

Statistical and AI (artificial intelligence) approaches [4] are mostly used to predict time series. Statistical methods can predict time series well but not so well when there are noises in the time series such as inaccurate or incomplete data. On the other hand, neural networks are one of the AI methods. Neural networks use generalized regression to approximate nonlinear time series. Neural networks are applied for better prediction in comparison with statistical methods and become an alternative to classical methods to perform time series prediction. However neural networks have difficulty in catastrophic forgetting, excessive training time and lack of knowledge representation facilities. It's hard to justify the correctness and confidence of a neural network prediction because of its numerical representation. Results of neural network prediction depend on the neural network architecture and the length of time series. This is because the neural network uses numerical knowledge representation and this causes the user unable to know how well the neural network captures its knowledge.

I've decided for a new approach to predict time series in this work. The approach uses adaptive association rules for time series life long prediction [1]. I suppose recurrence of some motion patterns in our model. The degree of prediction for a model means how many points ahead a model can predict. The degree of this model is dynamic and it depends on concrete video. It uses previous data and so if a time series has high repetitive pattern rate, the degree of prediction will be long and if this rate is low, the degree of prediction will eventually be short.

This model should recognize random data because this model depends on repetitive patterns. Thus, if time series is random, repetitive patterns will be too low. This model provides confident level for each prediction according to previous data.

2.1 ADAPTIVE ASSOCIATION RULES

Association rules correlate the presence of a set of items with another range of values for another set of variables [1]. For examples, if an object moves repeatedly from left bottom to the right top corner, there is a probability that object which is in the left bottom corner will move to the right top corner.

This data mining model works in few steps. First step is pre-processing time series data. It means converting time series data into chain code so the time series can be processed by data mining. A times series is represented in a chain code format to keep track of the rate-of-change of a time series. Chain code representation of a time series is deduced from gradient between every two points in time series. Every chain code gradient is labeled. Each label consists of range of gradient value. We can sectionalize values of gradient into few categories, which can be named A, B, C,… This chain code will be processed into multiple sizes and then into multiple itemsets which will be processed by data mining. Data mining will find repeatable patterns from every itemsets. These patterns will be stored in a database (or XML document) in the association rules form. Figure 1 shows the process flow of generating the chain code.

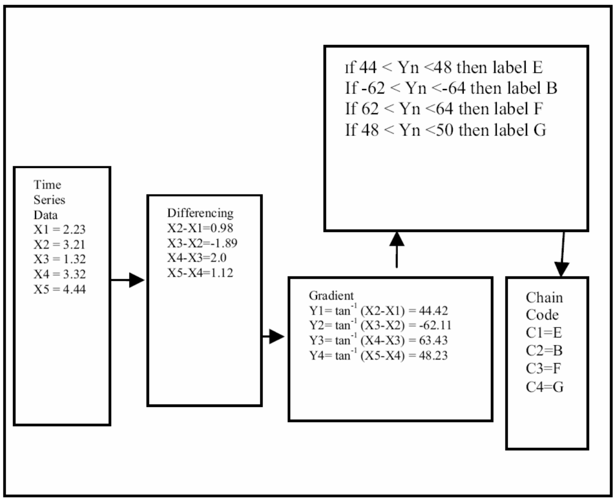

Fig. 1: *The process flow of generating the chain code.*

Next phase is application of pre-processing on the time series before performing prediction. The prediction is done by taking C(t –1), C(t – 2), C(t – 3) … C(t – n) where the C is the chain code value, t is the time and n is the number of points that being taken from the time series as input to perform pattern matching with the discovered repetitive patterns in the database.

If inputs match with stored repeatable time series pattern, we can predict following segment with the certain probability. This prediction model will then use the following segment as prediction and the possibility of reoccurrence as the confidence level. Number *n* is the number of points which are used to perform prediction. Its value is dynamic due to depending on the confidence threshold. There is a possibility to use more then one pattern to perform prediction. We can illustrate situation on a simple example. We have a pattern which uses *n*=12 with confidence 60 % and a pattern which uses *n*=5 with confidence 30%. Method will use the pattern with n=12. However, if there are more patterns with same *n* (for example *n*=9) in database, system will use all ones to perform prediction. Thus, the method is able to generate one or more predictions on the same time series.

Confidence of each pattern is given by its implication possibilities. For example confidence of K, J, P imply B is computed as support(K, J, P, B) / support(K, J, P). Support means how frequently a specific pattern occurs in a time series. If support is low, it means that percentage that K, J, P implies B is low (according to history of a time series). Analogously, if support is high, the percentage of implication will be high. According to the time series history it means that the pattern of K, J, P => B has been discovered before and has occured several times. So, the method will predict next segment value mostly probably going to be B and provide certainty of reoccurrence B as support(K, J, P, B) / support(K, J, P). This process repeats for the entire repetitive pattern that was discovered earlier by the data mining process. For example, we have found a pattern E, B, F, G, H earlier in time series. We will calculate the certainty of E, B, F, G, H to reoccur if E, B, F happens.

Data mining process is usually single session process. It means that this process is usually executed once over all data. Data mining process won't be executed again and only output is processed further. So the method is modified to be a lifelong process. It provides us better adaptation and reinforcement learning. In order not to have to repeatedly pass through time series, the support value of each pattern is modified. If the prediction of the model is incorrect, the pattern support value is decreased. On the other hand, if the prediction is correct, the support value is increased. So we have stored only support values and not confidence. For example, let support (E, B, F, G) = 6 and support (E, B, F) = 10. Confidence that G is going reoccur will be calculated as support (E, B, F, G) / support (E, B, F) which is 0.6. If prediction is correct, support values will be increased ((E, B, F, G) = 7 and support (E, B, F) = 11) so value of confidence will be 0,63. On the other hand if prediction is wrong, only value of support of (E, B, F) will be modified (support (E, B, F, G) = 6 and support (E, B, F) = 11), it will reduce value of confidence to 0.54. At the same time, new pattern will emerge if the prediction is wrong such as support (E, B, F, X) = 1 will be keep in database. If support (E, B, F, X) keeps on increasing, it will eventually have higher confidence level that X is going to reoccur instead of G. Therefore, the model will enable to continuously take input from a time series and continuously mines for new patterns and discards patterns that no longer repeat by modifying those repetitive patterns' support in the database

3 XPATH QUERIES OVER XML DATAS

There are some methods for effective and simple extracting data from XML documents (e.g. XQuery, XPath). XPath will be applied in this project. XPath provides a common syntax and semantics for functionality shared between XSL Transformations (XSLT) and XPointer. XPath was primary developed for addressing of elements in XML documents. So it also provides basic facilities for manipulation with basis data types (strings, numbers, booleans). XPath uses non-XML syntax for easy usage of XPath within URIs and XML attribute values.

Input data generated by M4 project is stored in XML document. There will be presented a piece of input XML document which will be used for getting data.


```xml
<?xml version="1.0"?>
<AVEvents>
<Events>
   <Name ID="0">Head</Name>
  </Events>
  <File>
   <Source Camera="2">Cam2_T000010.080_T000506.920.avi</Source>
   <TimeFormat>Frames</TimeFormat>
   <Event>
     <ID>0</ID>
     <Time>0</Time>
     <Parameters Camera="2" Object="13" CenterX="191,805246137262"
CenterY="466,166726554078" MinX="140" MinY="440" MaxX="243" MaxY="488"
Person="A"/>
   </Event>
  </File>
</AVEvents>
```


So if we want to predict motion of the head, we need to get all XML nodes named Event identified by <ID>0</ID>. We can get all these nodes with XPath query and then we can go through result and get some repetitive patterns. XPath query will be designed like this "AVEvents/File/Event[id = '0']". So we can design some function to get a values of position of the object Head in time. The main part of a function for extracting requested data can be designed something like this:

```
nodeList->selectNodes(BSTRWrap(toBSTR("“AVEvents/File/Event[id=  “0”]”)),&nodeList);
nodeList->nextNode(&node);                    //return collection of requested nodes
while(node!=NULL){
node->selectSingleNode(BSTRWrap(toBSTR("./time")),&nodeT);
nodeT->get_text(&dataTime);                   //return value of id element
node->selectSingleNode(BSTRWrap(toBSTR("./parameters/@CenterX")),&nodeT);
nodeT->get_text(&dataX);                      // return value of CenterX attribute
node->selectSingleNode(BSTRWrap(toBSTR("./parameters/@CenterY")),&nodeT);
nodeT->get_text(&dataY);
index=list->InsertItem(index, dataTime, dataX, dataY);     // storing of extracted data
index++;
nodeList->nextNode(&node);                    // go to next node
}
```

With previously algorithm we get time series data. This data will be processed by data mining. We can also store all intermediate results in XML data during data mining process, so we can design system which works only on XML data with no need to use some database.

4 CONCLUSION AND FUTURE RESEARCH

In this paper was presented method for object motion prediction. This method is a follow-up to a method for a time series prediction. Time series is explored to get repetitive patterns which provide us a possibility of prediction future state missing data sequences.

In my future work, I will test this method on real data. I will also study how motion vector can be processed without need of separating on horizontal and vertical components. If this method will be successful in single object prediction, it will be improved to predict motion of groups of objects such a combination of parts of human bodies or combination of more then one person in video.

REFERENCES

[1] Yaik, O. B., Yong, C. H., Haron, F.: Time Series Prediction using Adaptive Association Rules.First International Conference on Distributed Frameworks for Multimedia Applications (DFMA'05),p. 310-314, Besançon, France, 2005.

[2] Braga, D., Campi, A., Ceri, S., K1emettinen, M., Lanzi, P. L.: Discovering interesting information in XML data with association rules. Proceedings of the 2003 ACM symposium on Applied computing, p. 450-454, Melbourne, Florida, 2003, ISBN:1-58113-624-2.

[3] Zhang, S., Zhang, J., Liu, H., Wang, W.: XAR-Miner: Efficient Association Rules Mining for XML Data. Special interest tracks and posters of the 14th international conference on World Wide Web, Chiba, Japan, 2005, ISBN:1-59593-051-5.

[4] Glymour, C., Madigan, D., Pregibon, D., Smyth, P.: Statistical Inference and Data Mining. Communications of the ACM archive, Volume 39, Issue 11, (November 1996), p. 35 - 41, New York, NY, USA, 1996, ISSN:0001-0782.

[5] Renals, S.: M4 - multimodal meeting manager [Online]. http://www.m4project.org/.