Mining Time Series Using Rough Sets - A Case Study

Anders Torvill Bjorvand
Troll Data Inc.
P.O. Box 335
N-1801 Askim, Norway
torvill@trolldata.no

Abstract

This article attempts to deal with the problem of time within the framework of rough sets. The rough set theory has emphasized the reduction of information necessary to acquire desired knowledge. This is particularly important when we are dealing with time. The farther back we are tracing our dependencies, the more attributes will become independent of our current decisions.

We formalize approaches to reasoning with time series where the sequence of events is important, and introduce formalisms to deduce decision rules with real-time constraints.

1 Introduction

One specific problem relating to many interacting events, is the problem of time dependencies. Traditional rough set theory has emphasized the reduction of information necessary to acquire desired knowledge. With time series, the farther back we are tracing dependencies, the more attributes will become independent of our current decisions. If we can find sequences in time both in single attribute changes and in changes between attributes, we can find better, shorter rules.

There are many areas where there is a need to make predictions and classifications based on time series such as eg the stock market, patient medical history and control systems.

Therefore - in this paper, I will focus on objects and attributes that are dependent on each other in time. I will suggest a framework that extends the traditional information system of rough sets to include a sequence/order of events. An outline showing how these theories might be extended to deal with real-time constraints will also be given.

Some experiments have been carried out on monthly stock market data to

support the theoretical work.

General knowledge of rough set theory is required - cf eg [9] or [10].

1.1 Related Research

Temporal Reasoning

Much work has been carried out with mathematical models based on e.g. Fourier series and probability distributions and correlations (see eg [7]).

Interesting paths of research have also been taken in the area of temporal logic. There have been attempts to model real-time sequences with temporal logic. This was suggested by Jonathan Ostroff in [8], where he introduced the framework of Real-time Temporal Logic (RTTL). Within this framework, he used an event variable, n, to refer to events, and a clock variable, t, to refer to the clock value. This gave him the opportunity to state more than just the proper sequence of events.

Temporal Reasoning With Rough Sets

One approach is reported in [1]. Market data from the Hughes Research Laboratory have been analyzed, and some success in predictions has been reported. Dynamic reducts are utilized, and have been reported as a necessity in this application. The data used variants of each indicator representing both the present level and previous trends.

Another approach is reported in [5] and [6]. They have based their analysis on Canadian stock market data on a monthly basis. No predictions are reported quantitatively however. The translation from a time series to a traditional information system is introduced in this paper.

Time varying information systems have also been discussed in [11].

2 Reasoning With Time Sequences

For illustrational purposes, I shall use a small subset of the stock market data experimented with in section 4.1.

Example 1 *We have used discretized, monthly values of one stock value and two economic indicators. Each value represents the percent change since last month. -1 represents a decrease of 0% to 5 %. 0 represents an increase of 0% to 5%. 1 represents an increase of 5% to 10%.*

Month	*Dow Jones*	*Standard and Poors Index*	*Decision: Northern Telecom*
1	*0*	*0*	*1*
2	*-1*	*0*	*-1*
3	*-1*	*-1*	*-1*
4	*-1*	*0*	*-1*
5	*0*	*0*	*1*
6	*0*	*0*	*-1*

To deduce dependencies and make predictions based on sequences, the attribute 'Month' in our example is very important, and we must know the correct sequence of the objects. This calls for a slight modification of the information system. We introduce the temporal information system so that we can formalize information about the sequence/order of the objects.

2.1 Temporal Information System

Definition 1 ***Temporal Information System (TIS)*** $\mathbf{A}_t = (U, A \cup \{d, t\}, \prec)$
U— objects (cases, states, patients, observations,....)
A-features, variables, characteristic conditions,.....
d - the decision attribute : $d \notin A$
t - the sequence attribute : $t \notin A$
$\prec$ is an ordering relation on the sequence attribute, t.
$\prec = \{(x, y) : x, y \in \mathbb{N} \text{ and } x < y\}$

Having formalized the sequence of our objects, we are ready to move on. Defining a limit as to how far back dependencies/sequences should be traced, we can translate a TIS into an IS. This is desirable because of the formalisms already developed and the computational techniques available to us through the IS representation.

2.2 How to Translate a TIS Into an IS

Using a TIS and traditional RS techniques, we cannot deduce any temporal relations from our small stock market example. To do this, we need to translate the TIS into an IS where data from different periods can be investigated simultaneously. I will show how this translation of the TIS of example 1 can be carried out:

Example 2 *In this example, we set a limit of 2 months for backward dependencies. We then get the following results:*

Dow Jones Change this month	*Standard and Poors Index Change this month*	*Dow Jones Change last month*	*Standard and Poors Index Change last month*	*Decision: Northern Telecom*
-1	*0*	*0*	*0*	*-1*
-1	*-1*	*-1*	*0*	*-1*
-1	*0*	*-1*	*-1*	*-1*
0	*0*	*-1*	*0*	*1*
0	*0*	*0*	*0*	*-1*

Going back two months, we will lose one object, since we have no month previous to month 1. To put it more simply, we keep this month's decision attribute, and align all the conditional attributes from as many months as we wish to go back.

This procedure was introduced in [5]. We will now attempt to formalize this algorithm.

We assume that we have n different values of the sequence attribute, and we assume that they are integer values from 1 to n. The number of objects would often be $\geq n$, but this algorithm assumes that $|U| = n$. This could be avoided by a more advanced algorithm, but we want to introduce a simple algorithm at this point.

Algorithm TIS to IS
Input : a temporal information system, $\mathbf{A}_t = (U, A \cup \{d, t\}, \prec)$ and a limit, Δ, defining how far back dependencies will be shown.
Output : a regular decision table $\mathbf{A} = (\hat{U}, \hat{A} \cup \{\hat{d}\})$
Procedure :

for all $a \in A$:
    assign a unique index between 1 and $|A|$ **so that** $A = \{a_1, a_2, ..., a_{|A|}\}$
$\hat{A} \leftarrow \varnothing$
for i from 1 to Δ :
    $B \leftarrow A$
    Concatenate i to the existing index so that $B = \{a_{i1}, a_{i2}, ..., a_{i|A|}\}$
    $\hat{A} \leftarrow \hat{A} \cup B$
$\hat{U} \leftarrow \varnothing$
for each $x \in U$ **such that** $t(x) > \Delta$:
    $\hat{U} \leftarrow \hat{U} \cup \{y\}$, where y has the following properties:
        $d(y) = d(x)$
        for each $a_{vw} \in \hat{A}$:
            $a_{vw}(y) = a_w(z)$ **where** $z \in U$ **and** $a_w \in A$ **and** $t(z) = t(x) - v$
One would also benefit from making the data available in a percent-change format.
If you are trying to find dependencies in eg stock market data, you will never see exactly the same sequence twice. What might repeat itself, however, is the pattern of change: down $\rightarrow$ down-strong $\rightarrow$ no change $\rightarrow$ up. So, in many practical applications, it is beneficial to reason about the patterns/sequences of change rather than the direct periodic measurements. This could be done through preprocessing before converting from a TIS to an IS:

Algorithm Preprocessing for absolute change in a TIS
Input : a temporal information system, $\mathbf{A}_t = (U, A \cup \{d, t\}, \prec)$
Output : a temporal information system $\hat{\mathbf{A}}_t = (\hat{U}, \hat{A} \cup \{\hat{d}, t\}, \prec)$
Procedure :

for all $x \in \hat{U}$:
    for all $a \in A$ **we have a corresponding** $\hat{a} \in \hat{A}$ **where** :
        $\hat{a}(x) = a(y_2) - a(y_1)$ **where** $y_1 \in U$ **and** $y_2 \in U$ **and** $t(y_2) = t(x)$
and $t(y_1) < t(y_2)$ **and** $\neg\exists y_3((t(y_3) > t(y_1)) \wedge (t(y_3) < t(y_2)))$
In many cases it would be better yet to normalize the change using techniques for representing change like e.g. percent change and relative log change like the DeciBel measure.

2.3 Dealing With Different Sampling Rates in the Source Data

When we are dealing with different sampling rates in the source data, we are approaching a special case of the null value problem. Since we can have stock market data from different sources, some attributes may be on a monthly basis, but others may be on eg a quarterly basis. There are many solutions to this problem. We can try to get all our attributes on a quarterly basis, but this is rarely a good solution. The best way is to try and predict the value through different curve fitting strategies like linear descent. In [5], the quarterly values were simply repeated for each month. Fortunately - there are more sophisticated ways of "filling in the blanks".

2.4 Efficient Data Reduction by Utilizing the History

As it was mentioned in the introduction, rough set theory is particularly well suited for data reduction.
By increasing the number of attributes and keeping the number of objects constant (almost), we will increase the possibility of obtaining shorter reducts. This is exactly what is done with the translation from TIS to IS. We therefore propose the following theorem:

Theorem 1 *By utilizing the history of an information system (IS), we will have a good possibility of obtaining shorter reducts.*

This will be tested experimentally in section 4.

3 Reasoning With Time Sequences with Real-Time Constraints

Many applications have real time constraints. You have to know more than the sequences of events - you must also know the time between them. Eg when you are logging errors in a computer, many errors will trigger a hardware interrupt. These interrupts will come at irregular intervals.
To be able to reason while taking the irregular time intervals into consideration, we introduce a special version of the information system.

3.1 Real-Time Temporal Information System

This representation could store different sequences in a computer network, the different operational steps performed by a manual operator for controlling an industrial process. The difference from the TIS is that we also store the time interval between each event in the sequence.

Definition 2 ***Real-Time Temporal Information System (RTTIS)*** $\mathbf{A}_{rt} = (U, A \cup \{d, t, \delta\}, \prec)$
U – objects (cases, states, patients, observations,....)
A-features, variables, characteristic conditions,.....
d - the decision attribute : $d \notin A$

t - the sequence attribute : $t \notin A$
$\prec$ is an ordering relation on the sequence attribute, t.
$\prec = \{(x, y) : x, y \in \mathbb{N}$ *and* $x < y\}$
δ - the time attribute : $\delta \notin A$
δ is the time since the previous object with respect to the sequence attribute:
So $\delta(x_2)$ is the time since the object x_1 happened, where $t(x_1) < t(x_2)$ and $\neg\exists y((y > x_1) \wedge (y < x_2))$ where x_1, x_2 and y are members of U.

3.2 How to Translate a RTTIS to a TIS

There are many possible approaches to translate a RTTIS to a TIS. I will only give a brief outline of the possible solutions to this problem.
The common theme for all solutions to this problem is that we have to obtain a uniform frequency in our data. In a RTTIS, data may have been logged at highly irregular intervals. This frequency can get very high in order not to lose data, so we might need a postprocessing step to lower the frequency a little.
By requiring δ to be an integer, the period of this frequency should be the greatest common divisor of all $\delta(x)$ for $x \in U$. With large amounts of data, this period will very often become 1. When we have obtained the common frequency, we should generate the TIS by producing an object for every period of time according to this common frequency. Different curve fitting strategies should be utilized for this step. One might discover that the number of objects is too large to handle in our new TIS. That may be resolved by examinig the data and lowering the frequency.

4 Experiments

I want to verify the theories and algorithms described and put forward in this paper. The data set consists of stock market data.

4.1 Stock Market Data

The data analyzed in this section are mostly from the Canadian stock market. They are time-series on a monthly basis over a ten year period.
A description of the data can be found in [12] or [6], and the data can be obtained through the Internet at ftp://ftp.cs.uregina.ca/pub/ebrsc.
Goal
The goal of the experiment was twofold: To make use of the preprocessing algorithm described in section 2.2 (TIS to IS). and to verify that we will get shorter reducts by tracing back in history.
Results
The preprocessing and reduct search was done with version 2.0 of the system Rough Enough described in [2], [4] and [3]. When I analyzed the data one month backward, the shortest reduct was 6 attributes. When I analyzed two months backward, the shortest reduct was 4 attributes. This is consistent with my theorem.

5 Conclusion

I have introduced several representations and algorithms for dealing with time series. One important result of my experiments was that the reduct for tracing two months back was shorter than the reduct for tracing one month back. This is consistent with my theorem and shows that the rough set theory is well equipped to handle reasoning with sequences, since the potential of data reduction is considerable. The transformation from TIS to IS is significant since it enables us to use standard RS techniques for further processing of the data.

6 Future Research

The attribute values in traditional Information Systems of rough set theory are constants. I believe it would be fruitful to investigate the possibility of replacing these with functions, $f(x_1, x_2, ..., x_n)$ of n parameters giving an information system of dimension n+2. My temporal information systems are special cases of these - with a function $f(t)$ resulting in three dimensions.
More directly related to the problem at hand is the problem of relating attribute transitions to time itself and not to the transition of other attributes. This is particularly important when working under real time constraints. A solution to this problem from the temporal logic point of view has been outlined in [8] within the framework of Real Time Temporal Logic (RTTL).

References

[1] Bazan, Jan G., Skowron, Andrzej, and Synak, Piotr (1994). *Market Data Analysis: A Rough Set Approach.* Technical report from the University of Warsaw.

[2] Bjorvand, Anders Torvill (1996). *Time Series and Rough Sets.* Master's Thesis, the Norwegian Institute of Technology, Department of Computer Systems, Trondheim, Norway.

[3] Bjorvand, Anders Torvill and Komorowski, Jan (1997). *Practical Applications of Genetic Algorithms for Efficient Reduct Computation.* 15th IMACS World Congress 1997 on Scientific Computation, Modelling and Applied Mathematics - to appear.

[4] Bjorvand, Anders Torvill (1997). *Rough Enough - Software Demonstration.* 15th IMACS World Comgress 1997 on Scientific Computation, Modelling and Applied Mathematics - to appear.

[5] Golan, Robert and Edwards, Donald (1993). "Temporal Rules Discovery using Datalogic/R+ with Stock Market Data". Printed in *Rough Sets, Fuzzy*

Sets and Knowledge Discovery. Edited by Ziarko, Wojciech P. From the *Workshops In Computing* series edited by van Rijsbergen, C. J. Pp. 74-81. Springer-Verlag.

[6] Golan, Robert and Ziarko, Wojciech (1995). A Methodology for Stock Market Analysis Using Rough Set Theory. Printed in Proceedings of IEEE/IAFE Conference on Computational Intelligence for Fiancial Engineering, New York City, pp. 32-40.

[7] Janacek, Gareth and Swift, Louis (1993). *Time Series - Forecasting, Simulation, Applications.* Ellis Horwood.

[8] Ostroff, Jonathan S. (1989). *Temporal Logic for Real-Time Systems.* Research Studies Press LTD, John Wiley & Sons Inc.

[9] Pawlak, Zdzisław (1991). *Rough Sets - Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, Dordrecht.

[10] Skowron, Andrzej (1995). "Synthesis of Adaptive Decision Systems from Experimental Data". *Printed in SCAI'95 - Proceedings of the Fifth Scandinavian Conference on Artificial Intelligence.* Edited by A. Aamodt and J. Komorowski. IOS Press.

[11] Zakowski, W. (1993). *Sequences of Information Systems, Configurations and Conflicts.* Bull. Pol. Acad. Sci. Ser., Tech., 41, 295-304.

[12] Ziarko, W., Golan, R. and Edwards, D. (1993). *An application of Datalogic/R Knowledge Discovery Tool to Identify Strong Predictive Rules In Stock Market Data.* AAAI-93 Workshop on Knowledge Discovery in Databases.