RECOGNITION OF TRANSITIONS BETWEEN DIFFERENT PHASES OF THE PRODUCT LIFE CYCLE

Anatoly Sukov

Riga Technical University, Department of Modelling and Simulation, 1 Kalku Street, Riga LV-1658, Latvia

Anatolijs.Sukovs@cs.rtu.lv

ABSTRACT

Management of the product life cycle and of the corresponding supply network largely depends on information in which specific phase of the life cycle one or another product is. Finding a phase of the product life cycle can be interpreted as recognition of transitions between phases of life of these products. This paper provides a formulation of the above mentioned task of recognition of transitions and presents the structured data mining system for solving that task. The developed system is based on the analysis of demand of historical products and on information about transitions between phases in those products. The paper describes necessary data pre-processing and transformation steps, whose aim is to create a possibility of discovering rules in those data. The created rule discovering framework does not need a complicated realization, because the rules themselves can be discovered by a well-known and available classifier.

Keywords: data pre-processing, product life cycle, rule induction, classification

1. INTRODUCTION

Any created product has a certain life cycle. The term “life cycle” is used to describe a period of product life from its introduction on the market to its withdrawal from the market. Life cycle can be described by different phases: traditional division assumes such phases like introduction, growth, maturity and decline (Kotler and Armstrong 2006).

For products with conditionally long life cycle, it is possible to make some simplification, and, from the viewpoint of the dynamics of demand changes, the above mentioned phases can be merged into three. The first phase corresponds to introduction and growth; it is gradual or headlong growing of demand value in each of subsequent periods of time. The second and the third phases are the same - maturity and decline (also known as end-of-life). From the viewpoint of the management it is important to know, in which particular phase the product is. One of applications of that knowledge is selection of the production planning policy for the particular phase (Merkuryev, Merkuryeva, Desmet, and Jacquet-Lagrèze 2007). For example, for the maturity phase in case of determined demand changing boundaries it is possible to apply cyclic planning (Campbell and Mabert 1991), whereas for the introduction and decline phase an individual planning is usually employed.

From the side of data mining (Han and Kamber 2006) information about demand of particular product is time series, in which demand value is, as a rule, represented by the month. If there are different phases of the product life cycle, then there are different periods, in which transitions between these phases occur. Correspondingly, the task of recognition of the current phase for particular product consists of recognition of transitions between different phases of the product life cycle. It should be noted that such classical methods like Boston matrix (Kotler and Armstrong 2006) are unlikely worth to use for analysis of separate products, because those methods simplify the situation too much and use generalized information about groups of products. In such way, the need for creating a stable model for recognition of transitions between phases for each separate product exists.

This paper proposes a data mining framework for creating a model for recognition of the above described transitions. The model is based on the available in an enterprise database about the demand on historical products. The term “historical products” is used to describe those products, which already have transition of interest, for example, from introduction to maturity phase. After certain pre-processing steps and specific transformation of historical data, it is possible to apply a classifier, which is based on rule induction. As a result, the discovered rules are used for recognition of transitions between different phases of the analyzed product.

Successive parts of the work are organized as follows. Section 2 provides a more detailed definition of the transitions recognition task, and indicates necessary conditions and data for solving that task. Section 3 describes necessary pre-processing steps and presents specification of transformation of historical data. Selection of the classifier, and also discussion on the parameter setting, which influence the result of classifier work, are shown in Section 4. The results of real data experiments are discussed in Section 5. Section 6 summarizes features of the proposed framework and outlines possible directions of future research.

2. PROBLEM STATEMENT

If we assume that there are three different phases in product life cycle, namely, introduction, maturity and end-of-life, then two transitions are possible. The first transition is from introduction phase to maturity phase, and the second – from maturity to the product's end-of-life. Assume there are two products, for one of which information about the point of that product transition from introduction phase to maturity phase is available. The second product is just starting its life on the market and is similar to the first product with regard to the relative demand (see Fig. 1). If we are guided, for example, by the similarity degree of those two products, we will most likely make a decision that for the second product the transition will occur at the same time as for the other.

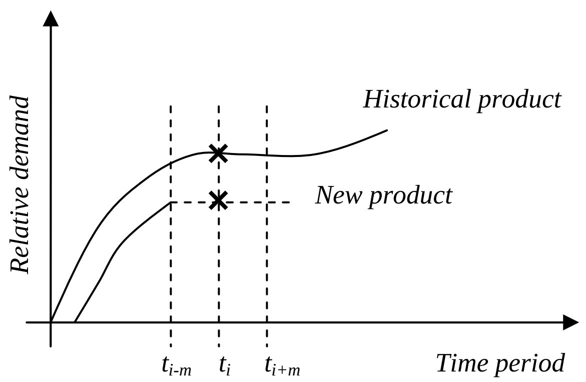

Figure 1: Three situations of transition time definition

The definition of the fact of transition can be conceptually described using three different techniques (see respective numbers in Fig. 1):

1. *Proactive approach* – we are at time moment t_{i-m} and wish to know the possibility that transition may occur at time t_i.
2. *Active approach* – we are at time t_i and wish to know the possibility that the transition is taking place right now.
3. *Reactive approach* – we are at time moment t_{i+m} and wish to know the possibility that the transition occurred at time t_i.

In a more general case, it is possible to predict (*proactive approach*), observe (*active approach*), or establish (*reactive approach*) the fact of transition both for one particular period, which corresponds to Fig. 1, and for a certain time interval.

From the point of view of practical evaluation of the above transition possibilities, it is evident that the proactive and active approaches will be characterised by higher uncertainty degree than the reactive approach. Parameter m must most probably be assumed to be equal to one, $m = 1$, which in combination with the reactive approach will enable creating an adequate model aimed to determine the fact of transition with the least uncertainty extent as compared to the two other approaches.

The situation with transition from maturity to the product's end-of-life is analysed similarly, which means that for such a transition the reactive approach is interpreted in the same way as for the transition from introduction to maturity.

2.1. Input historical data

The most practical and simplest case of available initial data is statistical data about the demand. Without doubt, in each period of time, in parallel with demand value, exogenous data exist, for example, orders and number of orderliness, number of customers for each product etc. Unfortunately, in practice it is difficult to obtain such complex information, therefore for the general case it is better to restrict data to the demand itself.

In general, the format of historical input data that could be processed by the system should (must) comply with these conditions:

- Each record displays the demand for a product, collected within known period of time, the length of which is set by the system – day, week, month, etc. In other words, each record is a demand time series.
- Each record has one or both markers – transition indicators:
  - Marker *K1* indicates the period when product switched from Introduction phase to Maturity phase;
  - Marker *K2* indicates the period when product switched from Maturity phase to End-of-Life phase.
- Each record has a marker, indicating the moment of the actual beginning of the Introduction phase (ABI).

The last condition is based on the fact that in many databases records are kept from the defined moment in time. It is evident that not all historical products were introduced on the market at the same moment in time. Marks on transitions can guarantee that a model will be build; if we have patterns of transitions in historical data, then, theoretically, in presence of a model for generalisation, we are able to recognise those patterns in new data.

3. MASTERING DATA: PRE-PROCESSING AND TRANSFORMATION

The main task of pre-processing demand data is to ensure that all time series are comparable. In the next turn, transformation is needed to remove excessive variety in the mentioned series, and also for representing initial task as a classification task. Practical utility of creating conditions for discovering generalities in time series is shown in the work (Das et al. 1998) where the authors propose to replace original time series with a sequence of discrete values, and after that to execute rule induction. The aim of the rule induction algorithm is to discover local regularities in that series. In case of recognition of transitions between phases, there is plenty of time series (that is why it is necessary to provide their comparability), and it is known that regularities should describe transitions exactly. Such specificity enables division into two classes – there is transition (*Yes*), there is no transition (*No*).

3.1. Pre-processing

Checking the fulfilment of the conditions mentioned in Section 2.1 is the first step in pre-processing historical input data. The next step in the pre-processing part is the shifting by *ABI*. The reason for this step to be included in the list is that *ABI* value varies between products. Shifting the records by the *ABI* corrects the variance of *ABI* between products by changing the period of actual beginning of Introduction to the first period. Together with *ABI* , the *K1* and the *K2* markers are shifted by the same number of periods as the *ABI* was – see the following formula (1).

$$\begin{aligned} K1' &= K1 - ABI + 1 \\ K2' &= K2 - ABI + 1 \end{aligned} \quad (1)$$

After $K1'$ and $K2'$ are calculated, we can assign $K1'$ value to $K1$ and $K2'$ value to $K2$. The next important step after the *ABI* shifting is completed, is to select the proper records for the learning set. The main predefined parameters for selecting records are the minimal *K1* and the maximal *K2* transition periods. The minimal *K1* parameter or $K1_{\min}$ defines the minimal Introduction to Maturity phase transition period allowed to be passed to the system. The maximal *K2* parameter or $K2_{\max}$ defines the maximal Maturity to End-of-Life phase transition period allowed to be passed to the system. It should be noted, that for ensuring adequate results $K1_{\min}$ should be fixed and equal to three periods, $K1_{\min} = 3$.

Different learning datasets are formed for processing *K1* transition and processing *K2* transition. Due to the defined parameters, records with $K1 < K1_{\min}$ and records with $K2 > K2_{\max}$ are marked as non proper records and are not selected. If a record has both *K1* and *K2* markers, but only one of the markers does not fulfil the defined conditions, then the record still can be marked as a proper record, but only one of the transition markers will be used. This means that if *K1* does not match the conditions but *K2* does, then this record still can be used in learning set for processing the *K2* transition, and vice versa.

To compare different series and to mine for the knowledge from a dataset, the data normalization is needed. It is recommended to use the *Z*-score normalization, as it uses the standard deviation to normalize data, and the demand interval bounds are not used (Han and Kamber 2006). Normalization of each time series is made separately.

The next step after the normalization of the input data is the selection of necessary amount of demand values. From each proper *i*-th record the necessary amount of demand values will be taken – for *K1* processing the $[L_{K1}, U_{K1}]_i$ interval and for *K2* processing the $[L_{K2}, U_{K2}]_i$ interval.

Calculating the interval $[L_{K1}, U_{K1}]_i$ of necessary periods we have:

$$L_{K1,i} = 1. \quad (3)$$

After shifting the records by ABI, the value of the $L_{K1,i}$ will be always equal to 1 – first period.

$$U_{K1,i} = \begin{cases} K1_i \textit{ is odd} : K1_i + 1 \\ K1_i \textit{ is even} : K1_i + 2 \end{cases}. \quad (4)$$

Calculating the interval $[L_{K2}, U_{K2}]_i$ of necessary periods we get:

$$L_{K2,i} = K1_i + 1. \quad (5)$$

Bound L_{K2} is the next period after the K1 switching period.

$$U_{K2,i} = \begin{cases} K2_i - L_{K2,i} + 1 \textit{ is odd} : K2_i + 1 \\ K2_i - L_{K2,i} + 1 \textit{ is even} : K2_i + 2 \end{cases}. \quad (6)$$

As can be seen from equations (4) and (6) the right bounds of intervals are selected in such a way, that: first, to guarantee that there will be one demand value after transition point, and, second, that common number of values will be even. Even length of the series is a necessary condition for following transformation of the data.

3.2. Data discretization

On the basis of pre-processed set of historical data the so-called set of blocks is created. Blocks are formed using non-overlapping sliding window of two periods. The amount of usually supplied demand data is less than 2 years (24 months, or periods). Due to that, the length of the block equal to 2 periods is the most suitable way for data generalization, as it lessens the loss of the data. The set of blocks is formed by extracting the blocks from each record in the learning dataset.

In order to made discretization, it is necessary to discover clusters in the set of blocks. For clusterization of blocks it is enough to use a simple partitioning algorithm, such as k-means (Jain, Murty, and Flynn

1999). Indications for choosing the number of clusters k are considered in detail in Section 4.1. After clusters are found, time series are replaced with a sequence of numbers of clusters, representing corresponding blocks of values. For simplification of the following usage, a symbol, for example, “C” is placed before the number of cluster. In such a way, those data will be nominal.

3.3. Simulating online data

Carefully selected for the analysis and discretized historical data, as a matter of fact, represents terminal situation – after a definite number of periods transition to another phase of particular product life occurs. In the time of analysis of new products data will come *incrementally*, by one period, and within each period the system should analyze demand series and report to the user if there is transition or there is no. Taking into account the specificity of discretization, the analysis will be made after each two new values. In any case this process will take place *online*.

In order to guarantee stability of the system work with *online* data, it is necessary to simulate such data in the learning dataset. This process also allows creating two classes of events, there is transition (*Yes*), or there is no transition (*No*). For example, if a record, which contains transition *K1* in the historical data, after pre-processing and discretization consists of three blocks, then in the learning this record will be used twice. First time of two first blocks (since $K1_{\min} = 3$, then minimal analyzed number of blocks is two) and in associations with class ”*No*”. And also this record will be used a second time with all three blocks and in associations with class ”*Yes*”.

3.4. Transforming data

The transformation of the data is needed to come up with records of equal length. A special symbol “C0” is used to mark blocks without data available, and to come up with records of equal length. The new target attribute (class) “Transition” is added. This is a binary attribute containing “No” for records without transition (simulated data), and “Yes” for record with switching period – original full record. Attribute “Blocks” contains the number of blocks with data available.

Since maximal number of possible periods is determined by parameters of used database, then from this number follows maximal number of blocks and dimensionality of data table obtained after transformation. For example, if, as it was mentioned above, the database is restricted by 24 periods, it means that in total the table will consist of 13 attributes and the class. An example is given in Table 1. Before transformation, the indicated record with ID=1 and ID=2 was one record, which consisted of three blocks – see an example in Section 3.3.

Table 1: Example of transformed data, K1 transition

ID	Blocks	B1	B2	B3	…	B12	Transition
1	2	C3	C1	C0	…	C0	No
2	3	C3	C1	C2	…	C0	Yes

4. RECOGNITION MODEL

The specificity of the dataset obtained for creation of classifier is such, that there is expressed a class disbalance. The fact that the data describes real life process and marks of transitions were putted by experts implies that some noisiness in data is present. Also it is evident that such dataset can have a large size – more than a thousand records after transformation, even if the initial database included historical information on a few hundred products.

Which of known classifiers can be successfully applied to the above-mentioned data? Taking into account a possibility of transparent interpretation of the result, rule induction was selected, in particular, the RIPPER algorithm (an acronym for *repeated incremental pruning to produce error reduction*) (Cohen 1995). Classes (“Yes” and “No”) are examined in the increasing order and an initial set of rules for the class is generated on one set, and then pruned on a separate data set. Each rule is pruned immediately after it has been grown – it is incremental reduced-error pruning. Having produced a rule set for the class, each rule is reconsidered and two variants produced, again using reduced-error pruning – but at this stage, records covered by other rules for the class are removed from the pruning set, and success rate on the remaining instances is used as the pruning criterion. If one of the two variants yields a better description length, it replaces the rule. A final check is made to ensure that each rule contributes to the reduction of description length, before proceeding to generate rules for the next class.

For carrying out practical experiments, realization of this algorithm in the Weka environment (Witten and Frank 2005) was chosen. The name of this classifier in Weka is JRip. The main parameters are the amount of data used for pruning (one fold is used for pruning, the rest for growing the rules), and the number of optimization runs. In the conducted experiments these default values were used: folds=3 and optimizations=2.

4.1. Setting the number of clusters

The major part of all possible parameters is setup by the data. A question then arises regarding the selected number of clusters for data discretization (see also Section 3.2). From a theoretical point of view, a smaller number of clusters provides a greater generalization of demand information, but the increasing number of clusters gives an opportunity to describe different transition situations in more detail. It is evident that the balance between generalization and detailed elaboration should be found for each database individually.

If we divide learning data into a number of subsets and conduct cross-validation with different number of clusters, then the exact number of clusters for particular dataset can be found. Naturally, for this purpose the data on which clusters are discovered, are not used for testing obtained rules. The criterion for selecting the necessary number of clusters could be the recognition rate of class “Yes” in the tested data. If there is a possibility of setting costs of misclassifications of both classes “Yes” and “No”, then total cost of misclassification becomes the criterion. Total number or errors cannot be used as a criterion because there is a disbalance between classes.

5. EXPERIMENTS WITH K1 TRANSITION

In order to show, which rules can be found by the JRip algorithm, experiments on a real dataset with K1 marks and corresponding transitions were conducted. The original dataset contained information on about 235 products. After pre-processing, 199 records remained, and they were divided into three equivalent size subsets. Thus, the threefold cross-validation was performed.

In total, after transformation, 199 representatives of class “Yes” and 769 representatives of class “No” were obtained. Experiment has shown that if the number of clusters is smaller than 4 and greater than 10, the recognition level of class “Yes” noticeably falls. The number of found rules varied from 5 to 10. Below an example of obtained rules is given (on the one of validation cycles) when the number of clusters was 7:

- (Block3 = C6) and (N_blocks >= 4) => Class=Y (31.0/9.0)
- (N_blocks >= 9) and (Block6 = C6) => Class=Y (22.0/5.0)
- (Block3 = C6) and (Block2 = C2) => Class=Y (16.0/6.0)
- (N_blocks >= 5) and (Block4 = C6) => Class=Y (30.0/9.0)
- (Block10 = C3) => Class=Y (24.0/11.0)
- (Block4 = C3) and (Block5 = C3) => Class=Y (5.0/0.0)
- (Block2 = C3) and (N_blocks >= 6) => Class=Y (5.0/0.0)
- (Block2 = C6) and (N_blocks >= 3) => Class=Y (12.0/4.0)
- => Class=N (502.0/32.0)

The first eight rules describe class “Yes”, and the last rule indicates class “No”. After each rule, it is shown how many examples it covers and how many of these examples were misclassified. In this particular case recognition of class “Yes” was 71%, but recognition of class “No” 93%. It is evident, that obtained rules are easy to interpret and to explain to the business-users.

Is the obtained result occasional? To make sure that functioning of the system is stable, from all records in the original dataset (Set1) first values of demand were removed. The obtained set (Set2) was similarly tested at different numbers of clusters k, from 4 to 9. The received results for both datasets are shown in Figure 2. The recognition level of both classes for different datasets varied within 5%, which indicates the stable work of the developed system.

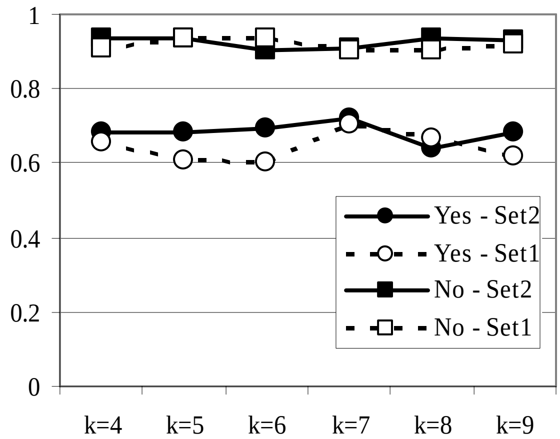

Figure 2: Recognition rate and used number of clusters

6. CONCLUSIONS AND FUTURE WORK DIRECTIONS

For the practitioners of management of the product life cycle the knowledge, which describes in which phase the product is, is topical. Such knowledge, in particular, helps to select between the cyclic and non-cyclic policy of planning supply chain operation.

In this paper, the task of recognition of transitions between different phases of product life is stated, and the structure of data mining system, which helps to solve this task, is shown. On the basis of the analysis of demand data on historical products it is possible to create stable classification rules, which are able to give online answers – if there is a transition in the particular product or there is no. The obtaining of such rules becomes possible thanks to creation of pre-processing and transformation system. The systems use demand time series as an input, and output strictly structured records. Since each obtained record has a class, it is possible to apply well- known and robust classifiers. From the point of view of implementation, the developed system is not complicated. All pre-processing and transformation processes are relatively simple, and it is possible to obtain rules in a well-implemented and available Weka environment.

One aspect is that, in the future it is necessary to examine the developed system on the data from different production fields, and, which is also important, to have a response from practitioners of supply chain management who will use these systems. In the near time described system will be implemented at a real enterprise.

Another aspect, modest data volume that was used for practical experiments, is related to the fact, that it is necessary to have transition marks in historical data from experts and practitioners. The more products, the more complicated for human to make all these marks – in practice the amount of marked data will always be restricted. As a result, possible direction of future research is treatment of recognition of transitions in the context of a semi-supervised learning (Zhu 2005). In this case, there is a small set with marked transitions (classes) and also a large dataset in which classes are not marked. In such a situation it is necessary to create a model, which will be able to recognize classes not only in the marked data, but also in the new (test) data.

ACKNOWLEDGMENTS

The present research was partly supported by the ECLIPS project of the European Commission "Extended collaborative integrated life cycle supply chain planning system", and also supported by ESF grant 2004/0002/VPD1/ESF/PIAA/04/NP/3.2.3.1/0001 /0002/0007.

REFERENCES

Campbell, G. M., and Mabert, V. A., 1991. Cyclical Schedules for Capacitated Lot Sizing with Dynamic Demands. *Management Science*, 37 (4), 409-427.

Cohen, W.W., 1995. Fast Effective Rule Induction. *Proceedings of the Twelfth International Conference on Machine Learning (ML95)*, pp. 115-123. (Tahoe City, CA).

Das, G., et al., 1998. Rule Discovery from Time Series. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 16-22. (New York, USA).

Han, J., and Kamber, M., 2006. *Data Mining: Concepts and Techniques*. 2nd ed. San Francisco: Morgan Kaufman.

Jain, A.K., Murty, M.N., and Flynn, P.J., 1999. Data clustering: A review. *ACM Computing Surveys*, 31 (3), 264-323.

Kotler, P. and Armstrong, G., 2006. *Principles of Marketing*. 11th ed. Prentice Hall.

Merkuryev, Y., Merkuryeva, G., Desmet, B., and Jacquet-Lagrèze, E., 2007. Integrating Analytical and Simulation Techniques in Multi-Echelon Cyclic Planning. *Proceedings of the First Asia International Conference on Modelling and Simulation (AMS 2007)*, p. 460-464. (Phuket, Thailand).

Witten, I.H., and Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd ed. Amsterdam: Morgan Kaufman.

Zhu, X., 2005 *Semi-supervised learning literature survey*. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Available from: http://pages.cs.wisc.edu/~jerryzhu/. [accessed on 10 February 2008]