PERGAMON

Expert Systems with Applications 21 (2001) 37–51

Expert Systems with Applications

www.elsevier.com/locate/eswa

A study on time series pattern extraction and processing for competitive intelligence support

Soe-Tsyr Yuan*, Ming-Zeng Huang

Information Management Department, Fu-Jen University, 510 Chung-Cheng Road, Hsin-Chuang, Taipei 24205, Taiwan

Accepted 27 February 2001

Abstract

In the era of rapid growth and high competition, a company must possess an information/knowledge advantage in order to hold the upper hand in the industry. Therefore, the company has to continuously monitor its competitors in order to get enough information and convert the information into competitive knowledge. Although information technology has been used in many areas and has many successful examples, it is rarely the case that information technology was used for the task of competitor intelligence. Accordingly, this study devises a method of time series pattern extraction and processing for the task of obtaining place-prospect competitor intelligence in order to advance an enterprise with a competitive knowledge advantage. The purposes of the method are two-fold: (1) For a product manufactured by the company, the gathered data are mined into a knowledge advantage—an appropriate amount of the stock to be allocated at a timely fashion at a retailer in face of competition. (2) This knowledge advantage alerts the company's decision makers to what is unknown and forces them to make good decisions on the stock allocation problem, freeing them from the dilemma of over-stock or under-stock with respect to competitors' stock. Our approach differs from traditional inventory management in the grounds they are based: traditional inventory management is based on the perspective of cash flow while our approach is based on the perspective of competition encountered. The results show our method is quite promising to this end, obtaining the intelligence to gain competitive advantage.

Keywords: Competitive intelligence; Time series pattern extraction

1. Introduction

It is a fairly accepted fact that competition will become even more intense in the 21st century compared to what it is today. As we move forward from the Information Age to the Intelligence Age, success will come to those companies that develop and maintain their competitive intelligence (CI) (see Competitor Analysis—a Brief Guide, hhttp://dspace.-dial.pipex.com/aware/competitor-analysis.html; Information on your competitors, hhttp://www.competitive-intelligence.co.uk/; Marketing Plan Components: Competitor and Issues Analysis, hhttp://www.onlinewbc.org/docs/marlet/mk_mplan_competitor.html).

CI is the process of obtaining vital information on your markets and competitors, analyzing the data and using this knowledge to formulate strategies to gain competitive advantage (Fayyad, Piatetsky-Shapiro & Smyth, 1996; Tyson, 1995). *This competitive advantage can be shown in different respects such as product, price, place, and promotion* (*4P in Marketing Management*). CI alerts you to what is unknown and forces you to make good decisions. This is different from market research which supports decisions that have already been made and mainly utilizes surveys or questionnaires for the understanding of the results of the strategic directions of their companies.

The rationale behind why CI is valuable to decision makers is described as follows (Tyson, 1995): (1) to maintain market share in the face of strong competition; (2) to identify opportunities for growth; and (3) to minimize threats. Key decision makers search for intelligence because it is information that can be acted on, resulting in a profoundly good effect on a company's market position and profitability.

In the past, companies relied heavily on CI professionals to help decision makers understand their market position and opportunities. However, human professionals have their limitations on the amount of information they can gather and analyze and the precision of the results they can obtain. To this end, *in this paper, for a particular real problem, a novel method of competitive intelligence support (CIMiner) is presented and evaluated, showing to the CI professionals its values mentioned above*.

* Corresponding author. Tel.: +886-2-2369-3220; fax: +886-2-2369-3220.

E-mail address: yuans@tpts1.seed.net.tw (S.-T. Yuan).

0957-4174/01/$ - see front matter
PII: S0957-4174(01)00025-2

The following is a description of the particular real problem this study is trying to conquer.

- For a given company (a worldwide leading printer company) that manufactures a set of products such as laser printers and ink-jet printers, it has numerous contracted retailers each of which is located at different place. For each of the products, at a specific time, each retailer has a particular amount of the product in stock (*Stock*) and a particular amount of the product sold (*Sale*). However, the retailer may also involve the retail of a similar product manufactured by a major competitor of the company and hence has a particular amount of this similar product in stock (*C-Stock*) and a particular amount of the similar product sold (*C-Sale*).
- The goal of CIMiner is to continuously provide intelligence for the decision makers of the company to act on for resulting in a good effect on a company's market position and profitability in face of a major competitor's threats.
- This intelligence is that for each product and retailer, *a suitable amount of the product that should be placed at a right time so that Stock can be in accord with Sale, and such an accordance/suitability depends on the major competitor*. In other words, this study aims to provide a *place-prospect competitive intelligence support system*, with which the company may rationally places its products at retailers to the benefits of the company in the long run.
- The benefits can be described in three-fold. (1) Minimize the cost of unnecessary holding inventory at the retailers (i.e. the situation as shown in Fig. 1(a) in which a great amount of inventory cost incurs when *Stock* overwhelms *Sale*). (2) Minimize chances of losing possible sales (i.e. the situation as shown in Fig. 1(b) in which there will be opportunity loss when *Sale* overwhelms *Stock*). (3) Maximize chances of having profits (i.e. the situation as shown in Fig. 1(c) in which there will be no unnecessary cost holding and no opportunity loss when *Sale* is in accord with *Stock*).

The difference between our problem and traditional inventory problems is described as follows.

1. Traditional inventory problems mainly concern about the amount of merchandise, parts, supplies, or other goods a business keeps on hand to meet the demands of its customers from the perspective of cash flow (Schreibfeder, 2000; and see Turnover Analysis, hhttp://www.advantageasessment/com/bibrary/TurnoverCost.htm). Depending on the nature of the business (i.e. retail, wholesale, service, manufacturing), the efficiency of inventory management may have a significant impact on the cash flow and, ultimately, the business's success or failure. For example, from a cash flow perspective, turnover analysis (see Turnover Analysis, hhttp://www.advantageasessment/com/bibrary/TurnoverCost.htm) was used for finding inventory items that are excessive, too low, or just right. (An excessive investment in inventory results in less cash available for other cash outflow purposes, such as paying bills.)
2. The problem addressed in this paper concerns the amount of a product required at a time at a retailer for a business of a printer manufacture from the perspective of competition encountered. That is, the information of the markets and the competitors is analyzed for obtaining the knowledge, which is unknown but good for decision making to gain competitive advantage.[1]

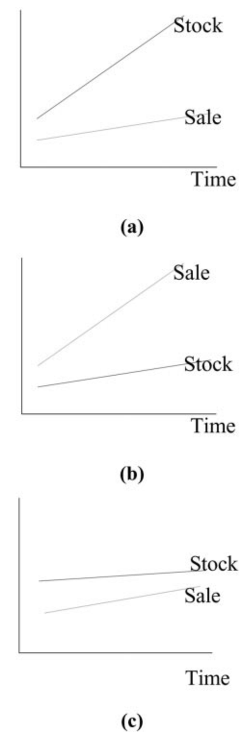

Fig. 1. The relations between Stock and Sale: (a) overstock (b) opportunity loss (c) accordant.

The method of the CIMiner is a combination of time series pattern extraction and time series pattern processing. The basis for the choice of this time-series-based approach is two-fold. (1) The amount of a product required at a retailer is time dependent. That is, suitable amount of the product that should be placed at a retailer may vary at different time in order to have Stock in accord with Sale all the time. (2) A natural way to come up with a time-dependent solution of the amount of the product required is using time-series-based approaches, which have been used to study patterns in

[1] Inventory management from both the perspective of cash flow and the perspective of competition encountered is worthy of future research.

data in order to understand the underlying processes and to predict their future behavior.

The rest of this paper is divided into five sections. Section 2 describes the data on which the CIMiner conducts experiments and the architecture of the CIMiner. Section 3 presents our method of time series pattern extraction. Section 4 then presents the method of time series pattern processing. Section 5 shows the evaluation results. Finally, a conclusion is made in Section 6.

2. Competitive information and the architecture of the CIMiner

Since CIs first task is obtain vital information on the markets and the competitors, in this section we first describe the information that the CIMiner uses for gaining a competitive advantage. Afterwards, the architecture of the CIMiner is presented.

When considering only one product for the company, CIMiner gathers the following sequences of information, such as time, retailer location, *Sale*, *Stock*, *C-Sale*, and *C-Stock*, and then cleans and integrates this data into a database as shown in Table 1(a). (The printer manufacturer our study working for regularly collected these data at retailers and kept them in different files as shown in Table 1(b).)

The data we have collected ranges in time between September, 1998 and February, 1999. The data involves 10 retailers and 360 records.

The process in the CIMiner that analyzes the integrated data and obtain competitive knowledge is two-fold: time series pattern extraction and time series pattern processing as shown in Fig. 2.

For each product and retailer, time series pattern extraction aims to discover problematic segments in which *Sale* is not in accord with *Stock*, and time series pattern processing subsequently provides a prediction of the suitable amount of the product needed to be placed in the future for possessing place-prospect competitive advantage.

Table 1
(a) Shows a fragment of the integrated data after CIMiner's gathering and preprocessing, and (b) shows the original data before CIMiner's preprocessing

Time	Retailer location	Stock	C-Stock	Sale	C-Sale
(a)					
2.1.1999	順發台中	20	15	10	13
6.1.1999	順發高雄	30	21	6	9
12.1.1999	順發新竹	16	35	15	15
(b)					
2.1.1999	順發台中	20	15		
6.1.1999	順發高雄	30	21		
12.1.1999	順發新竹	16	35		
2.1.1999	順發台中			10	13
6.1.1999	順發高雄			6	9
12.1.1999	順發新竹			15	15

3. Time series pattern extraction in the CIMiner

In this section, we first describe the general framework for time series pattern extraction and then delineate a novel way of employing this framework in the CIMiner. Finally, our algorithms are then presented.

3.1. The general framework of time series pattern extraction

Time series data is important for many application domains ranging from financial to scientific applications. An important use of time series is to study patterns in them in order to understand the underlying processes and to predict their future behavior (Bollobas, Das & Gunopulos, 1997; Das, Gunopulos & Mannila, 1997;Han, Dong & Yin, 1999; Li, Yu & Castelli, 1998; Qu, Wang & Wang, 1998).

A time series is assumed as an observation of some underlying process at an equal interval of time. Formally, given a positive integer n, a time series S of length n is a sequence of n real values, $V1$, $V2$, …, Vn. We use $S[i]$ to denote the ith value Vi of S, for $1 <= i <= n$.

Li et al. (1998)presented a framework (as shown in Fig. 3) for multiple abstract level analysis on time series data to discover complex search targets involving multiple levels of abstractions and details. In this framework, representations at multiple abstraction levels can be generated from the same time series, such as raw data level, feature level, and semantic level. The time series itself is at the lowest level. This framework can be described as follows.

- *Partition:* the time series is segmented into non-overlapping subsequences.
- *Extract features from each segment:* a linear approximation, such as standard regression techniques, to each segment can be applied and defines the characteristics of the subsequence. Examples of features are

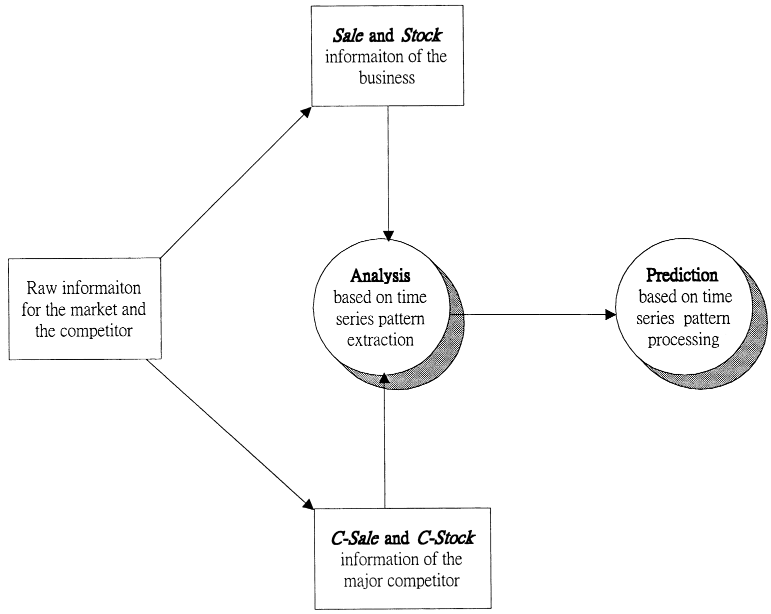

Fig. 2. The architecture of the CIMiner.

like the slopes of this linear regression, the mean square error and other statistics, constituting a feature level representation of the data.

- *Assign a label to each segment:* a labeling method, such as a simple heuristic or backpropagation neural network, to the extracted features of each segment can be applied to produce a semantic representation of the data, such as the vocabulary of rise, flat, and fall.

It is worth noting that by applying different segmentation, one can generate different representation of the same time series, resulting in different multiple abstract level analysis.

3.2. A Novel way of deploying the framework

One of the goals of the CIMiner is to discover problematic segments in which *Sale* is not in accordance with *Stock*, for each product and retailer. In time series pattern extraction, we have to identify the time series on which

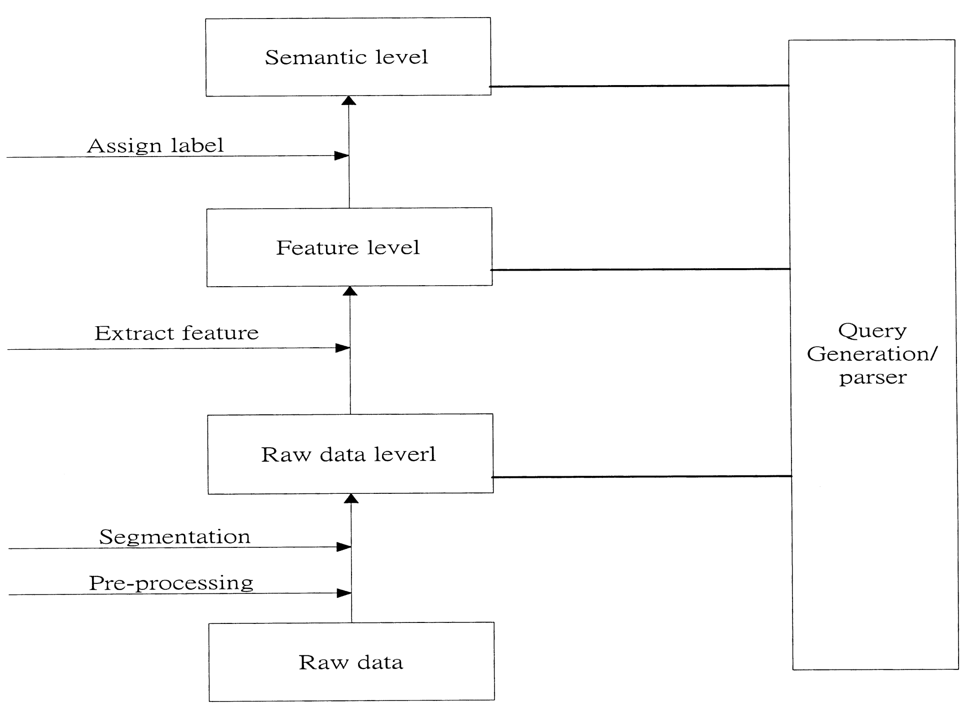

Fig. 3. The general framework for time series pattern extraction.

the analysis is conducted. In the past, a time series is simply an observation of some underlying process at an equal interval of time. *However, in our problem there are four such series, Sale series, Stock series, C-Sale series, C-Stock series, and they have to be related in some way in order to obtain the problematic segments*. It naturally poses a question—can we directly apply time series pattern extraction on these four series and then investigate the inter-relationship between the patterns of the four series in order to obtain the problematic segments? Or, alternative combined time series defined out of the four series can be investigated directly for finding the problematic segments.

In this paper, we take the second option, that is, we define alternative time series for the discovery of problematic segments. The rationale behind this choice is an intuition that each individual time series acts like a dynamic energy flow, and it is harder to analyze the interaction patterns between the four energy flows at once than directly analyze the interaction patterns of the combined energy flows.

As follows are the definitions of alternative time serieses the CIMiner uses, given a product and a retailer, for the discovery of the problematic segments.

- Stock_Ratio series: the observation of $Stock\_Ratio = Stock/(Stock + C - Stock)$ at an equal interval of time.
- Sale_Ratio series: the observation of $Sale\_Ratio = Sale/(Sale + C - Sale)$ at an equal interval of time.

In the definitions of the Sale_Ratio series and the Stock_Ratio series, the concept of *relative strength* is used in order to define the combined energy flow of *Sale* and *C-Sale* and the combined energy flow of *Stock* and *C-stock*. Subsequent analysis on the interaction of the patterns of the two combined energy flows can then be made. The concept of relative strength intends to capture the idea that the accordance between *Sale* and *Stock* should be dependent on the major competitor as mentioned in Section 1. Therefore, *afterward, by raw-level time series, we mean the Sale_Ratio series and the Stock_Ratio.*

In deploying the framework of time series pattern extraction mentioned in Section 3.2 (partition, feature level, and semantic level), we run into some decisions to be made: the segment length (Unit Length) with which a time series is partitioned, the features with which the characteristic of a segments can be identified, and the labels with which the semantic representation of a segment characterized by the features can be represented. The decision of Unit Length will be discussed in Section 5. We will devote the rest of this subsection to the descriptions of the features and the labels we use.

As follows are the definitions of the features CIMiner uses for given a Sale_Ratio series and a Stock_Ratio series.

- *Increment_Stock_Ratio*

$$= \frac{(\sum(Time2(Stock\_Ratio) - Time1(Stock\_Ratio)))}{(Data\_Num - 1)},$$

for the Stock_Ratio series.

Increment_Stock_Ratio: average increment between consecutive values of the sequence of Stock_Ratio values
$Time2(Stock\_Ratio) - Time1(Stock\_Ratio)$: the difference between two consecutive values of the sequence of Stock_Ratio values
$Data\_Num - 1$: the number of values in the subsequence (segment)

- *Increment_Sale_Ratio*

$$= \frac{(\sum(Time2(Sale\_Ratio) - Time1(Sale\_Ratio)))}{(Data\_Num - 1)}$$

Increment_Sale_Ratio: average increment between consecutive values of the sequence of Sale_Ratio values
$Time2(Sale\_Ratio) - Time1(Sale\_Ratio)$: the difference between two consecutive values of the sequence of Sale_Ratio values
$Data\_Num - 1$: the number of values in the subsequence (segment)

The feature of Increment_Stock_Ratio/Increment_Sale_Ratio intends to capture the trend of flow deviation for a segment of the Stock_Ratio/Sale_Ratio series. With the feature Increment_Stock_ratio for the Stock_Ratio series and the feature Increment_Sale_Ratio for the Sale_Ratio series, *we define a set of semantic labels that represent the degree of the accordance between the values of Increment_Stock_Ratio and Increment_Sale_Ratio.*

As follows are the semantic labels we have defined:

- Symbol ‘S’: ‘Safe’ segment if |Increment_Stock_Ratio − Increment_Sale_Ratio| < T
- Symbol ‘O’: ‘Overstock’ segment if (Increment_Stock_Ratio − Increment_Sale_Ratio) > T
- Symbol ‘L’: ‘Lack’ segment if (Increment_Sale_Ratio − Increment_Stock_Ratio) > T

T: a threshold for determining if the difference between Increment_Stock_Ratio and Increment_Sale_Ratio is acceptable.

The way we define the semantic labels is different from previous applications (Agrawal, Psaila, Wimmers & Zait, 1995; Das et al., 1997; Faloutsos & Lin, 1995; Li et al., 1998; Shatkay & Zdonik, 1996) of the framework of time

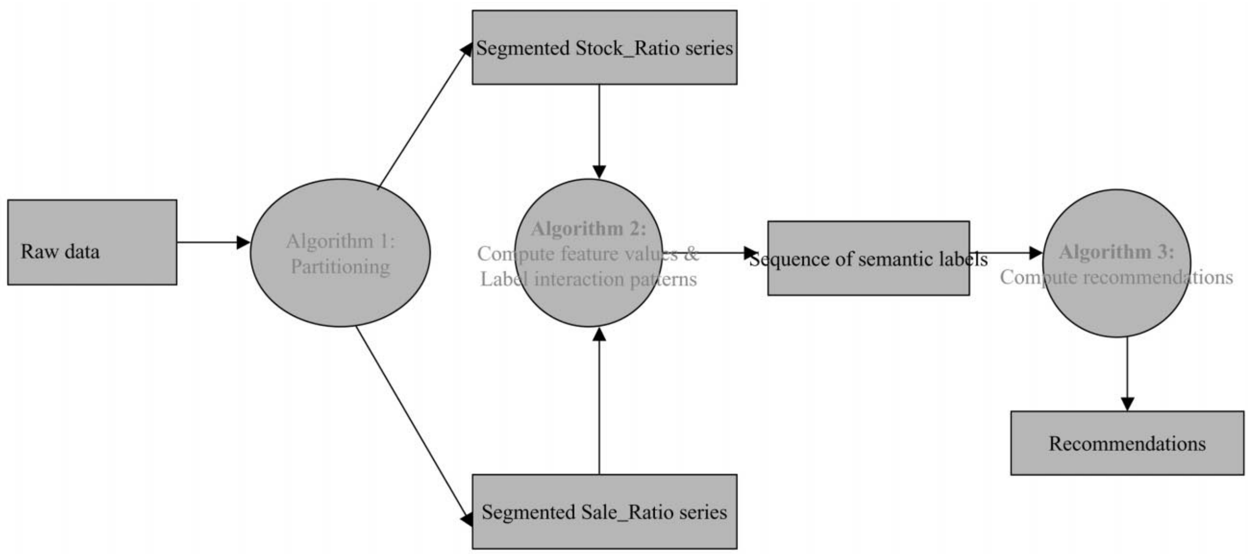

Fig. 4. The process flow between the algorithms.

series pattern extraction in the following ways:

- *Previous approaches defined the semantic labels out of the features of one segment in a given time series, while our approach defines the semantic labels from the features coming from two different segments in different time series.*
- *These semantic labels are used to represent the interaction patterns between the Stock_Ratio series and the Sale_Ratio series.*

3.3. The Algorithms

In this section, we present three algorithms: Algorithm1 and Algorithm 2 perform the task of time series pattern extraction, and Algorithm 3 performs the task of discovering problematic segments and time series pattern processing (recommendation provision). The details of time series pattern processing will be described in Section 4. The process flow of these three algorithms is shown in Fig.4.

With the raw data of the Stock_Ratio series and the Sale_Ratio series, Algorithm 1 performs the tasks of partitioning the series with the UnitLength and computing some statistics to be used in the task of time series pattern processing, resulting in two partitioned time serieses. Algorithm 2 then computes the values of the feature Increment_Stock_Ratio for the segments in the Stock_Ratio series and the values of the feature Increment_Sale_Ratio for the segments in the Sale_Ratio series and semantically labels the interaction patterns of the two serieses with the labels ‘S’/‘O’/‘L’ (characterizing segments with the semantic label ‘S’ if *Sale* is in accord with *Stock* and ‘O’/‘L’ if *Sale* is not in accord with *Stock* such as the situations of overstock and lack), resulting in a sequence of semantic labels. With the sequence of semantic labels, Algorithm 3 then is able to identify the problematic segments.

For each semantic label, Algorithm 3 identifies the maximum period in which a semantic label dominates the rest of semantic labels. By domination, we mean the frequency of the appearance of the semantic label is over a pre-specified threshold—80%. By a maximum period, we mean the problematic segments the CIMiner locates are as broad as possibly to prevent from the situation in which there are many fragmented problematic segments awaiting for correction. In other words, Algorithm 3 is able to identify maximum periods of problematic segments with semantic labels of ‘O’/ ‘L’.

3.3.1. Algorithm 1

Given: The raw sequences of sale data and stock data of the enterprise and its main competitor for a particular time period, and Unit_Length as the segment size
Algorithm 1:
  S_Date, E_Date//The beginning and ending day of a segment;
  Obj_Stock, Obj_Sale//Objects for storing stock information and sale information, for the segment period
  While (not the end of the period)
    Partition the series of unassigned data of length Unit − Length into a segment, for each raw sequence of data
    Create an Obj_Stock and an Obj_Sale
    Compute the values of Stock_Ratio in the segment period

Compute the values of Sale_Ratio in the segment period
Setup S_Date and E_Date in the Obj_Stock and the Obj_Sale
For I = 1 to Data_Num in the segment {
Sum_Stock_Ratio = the sum of Stock_Ratio in the segment
Sum_Sale_Ratio = the sum of Sale_Ratio in the segment
}
Setup Avg_Stock_Ratio in Obj_Stock by Avg_Sock_Ratio = Sum_Stock_Ratio/Data_Num
Setup Avg_Sale_Ratio in Obj_Sale by Avg_Sale_Ratio = Sum_Sale_Ratio/Data_Num
}
Return the sequence of Obj_Stocks
Return the sequence of Obj_Sales

3.3.2. Algorithm 2

Given: Two sequences of objects, Obj_Stock and Obj_Sale, for a particular time period, T as the Critical value, and Sequence as an empty combined series
Algorithm 2:
For (each consecutive objects in both sequences of objects) {
Compute Increment_Stock_Ratio
Compute Increment_Sale_Ratio
Setup D = Increment_Stock_Ratio − Increment_-Sale_Ratio
Select Case D {
Case D < T: Sequence = Sequence + 'S'
Case D > T: Sequence = Sequence + 'O'
Case −D > T: Sequence = Sequence + 'L'
}
return the Sequence
}

3.3.3. Algorithm 3

Given: the Sequence out of Algorithm 2, P as the frequency threshold for testing a particular dominated typed symbol of the T_Sequence.
Algorithm 3:
T_Sequence_S, T_Sequence_E//The Starting and Ending day of T_Sequence
T_Sequence = Sequence
While T_Sequence_S < T_Sequence_E {
Setup Sum(I) = the number of type I symbols in the T_Sequence, I = S,O,L
Setup Sum = the total number of symbols in the T_Sequence
Setup P(I) = Sum(I)/Sum, I = S, O,L
Select Case P(I) {
Case P(I) > P
Use the Correction Model to setup the Recommend Stock
Return the period of T_Sequence_S to T_Sequence_E
Case Other
T_Sequence_S = T_Sequence_S − 1}
}

4. Time series pattern processing in the CIMiner

In this section, we provide the ways we correct the problematic segments, Lack and Overstock periods, for the provision of the suitable amounts of the product at a retailer. Finally, an example is used to demonstrate the effect of the correction.

4.1. Correction models

We treat the cases of a Lack period and an Overstock period differently based on the results of our experiments:

- Lack period: the correction model is as follows:

$$\text{Re}\mathit{commend\_Stock} = \mathit{Stock} + \mathit{Ave\_Stock}^{*}\mathit{Increment\_Sale\_Ratio}$$

Recommend_Stock: the amount of the stock recommended
Stock: original amount of the stock
Ave_Stock: the average amount of stock among the segments in the period
Increment_Sale_Ratio: average increment between consecutive values of the sequence of Sale_Ratio values

The case of a Lack period implies that the amount of the stock does not keep up with the amount of the sale, that is, the difference between Increment_Stock_Ratio and Increment_Sale_Ratio is beyond the threshold T mentioned in Section 3.2. In order to make the stock be in accordance with the sale with regards to the major competitor, the correction model makes an adjustment based on Increment_Sale_Ratio with which the amount of adjustment is the product of the average stock in the period and Increment_Sale_Ratio.

- Overstock period: the correction model is as follows:

$$\text{Re}\mathit{commend\_Stock} = \mathit{Stock} - \mathit{Ave\_Stock}^{*}\mathit{Diff}^{*}\mathit{Adjust}$$

Recommend_Stock: the amount of the stock recommended
Stock: original amount of the stock
Diff: the difference between Increment_Stock_Ratio

Table 2
The data before correction

Record no.	Dates	Place	Stock	Competitor's stock	Sale	Competitor's Sale
1	13.11.1998	順發台中	48	79	18	14
2	19.11.1998	順發台中	53	83	16	12
3	25.11.1998	順發台中	55	95	13	11
4	1.12.1998	順發台中	63	60	13	12

and Increment_Sale_Ratio
Adjust: an adjust parameter defined as follows:

$$Adjust = \frac{(Ave\_Stock + Competitor\_Ave\_Stock)}{(Ave\_Sale + Competitor\_Ave\_Sale)}$$

Ave_Stock: the average amount of the stock
Competitor_Ave_Stock: the average amount of the competitor's stock
Ave_Sale: the average amount of the sale
Competitor_Ave_Sale: the average amount of the competitor's sale

The case of an Overstock period implies Increment_Stock_Ratio is much larger than Increment_Sale_Ratio. In order to make the stock be in accordance with the sale with regards to the major competitor, that is, eliminating the problem of overstock, the correction model needs to cut down the Stock and subsequently makes an adjustment based on the difference between Increment_Stock_Ratio and Increment_Sale_Ratio with which the amount of adjustment is the product of the average stock in the period, Diff, and an adjustment parameter. The purpose of the adjustment parameter simply converts the increment ratio difference in Diff to the real amount difference in the stock adjustment.

4.2. An example

In order to demonstrate the correction models mentioned in Section 4.1, we use a simple example (as shown in Table 2) in which there are a small number of real records in a particular period of time for a specific retailer and a product and the threshold T is preset to be 4%.

As follows are the necessary steps for the demonstration of the effect of a correction model:

1. Compute the Stock_Ratio for each record.
   Record 1: Stock_Ratio = (48)/(48 + 79) = 37.80%
   Record 2: Stock_Ratio = (53)/(53 + 83) = 38.97%
   Record 3: Stock_Ratio = (55)/(55 + 95) = 36.67%
   Record 4: Stock_Ratio = (63)/(63 + 60) = 51.22%
2. Compute Increment_Stock_Ratio.
   [(38.97% − 37.80%) + (36.67% − 38.97%) + (51.22% − 36.67%)]/3 = 4.47%
3. Compute the Sale_Ratio for each record.
   Record 1: Sale_Ratio = (18)/(18 + 14) = 56.25%
   Record 2: Sale_Ratio = (16)/(16 + 12) = 57.14%
   Record 3: Sale_Ratio = (13)/(13 + 11) = 54.17%
   Record 4: Sale_Ratio = (13)/(13 + 12) = 52%
4. Compute Increment_Sale_Ratio.
   [(57.14% − 56.25%) + (54.17% − 57.14%) + (52% − 54.17%)]/3 = −1.42%
5. Compute Diff.
   Diff = |4.47% − (−1.42%)| = 5.89%
6. As the threshold T is 4% that is smaller than 5.89%, the segment is considered as an Overstock period. Therefore, the CIMiner applies the correction model for an Overstock period by computing Ave_Stock = 54.75 and Adjust = 4.81 with which the CIMner updates the stocks for Record 2, 3, 4 as follows: (Record 1 is not corrected because it is the base on which the remaining records are adjusted.)
   Record 2: 53 − (54.75*5.89%*4.81) = 38
   Record 3: 55 − (54.75*5.89%*4.81) = 40
   Record 4: 63 − (54.75*5.89%*4.81) = 48

After the adjustment using the correction model, we can apply the same steps to the new records and obtain the new Diff as being 3.63% which is now smaller than the threshold T.[2] In other words, the CIMiner is able to make predictions about the suitable amount of the stock that should be placed in time at specific retailer for a specific product, based on the processing of the patterns obtained by analyzing the past data, so that the stock can be in accordance with the sale. In the Appendix, there are a few screen dumps that give the readers a look and feel about the execution of the CIMiner.

[2] The data is reused for testing because we assume the patterns in the history will recur and the amount of the data we can collect is too limited to be separated into two samples for training and testing.

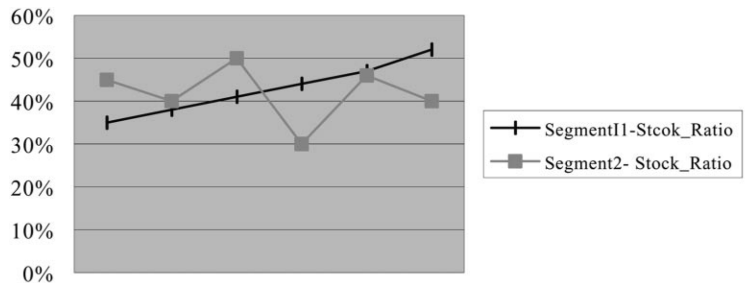

Fig. 5. An example showing the choices of UnitLength.

5. Evaluation

In the execution of the CIMiner, the task of time series of pattern extraction involves the decisions of the choice of UnitLength and the choice of the threshold T. It will be nice if there is a systematic way for determining these parameters. Accordingly, this section first explains the importance of finding good choices for these parameters and presents methods to determine these parameters. Finally, with the parameters set up by the methods, we can evaluate the quality of the correction models.

5.1. UnitLength

From our method of time series pattern extraction, we know different UnitLength would lead to different segments that further result in different preditions. Therefore, it naturally comes to a question—*on which grounds is a choice of UnitLengh based and said to be good or bad?*

Tracing back to the meaning of UnitLength, which is based on the flow trends, Increment_Stock_Ratio and Increment_Sale_Ratio are able to be defined. Therefore, *good choices of Unitlength are the ones that can really capture the flow trends in these two time series.*

The following are the metrics we use to evaluate a choice of UnitLength from the directions of Stock_Ratio and Sale_Ratio:

- The Stock_Ratio direction:

$M1' = |\text{Increment\_Stock\_Ratio}| * \text{Number of records in the segment}$
$M1 = \text{SUM}(M1')/\text{the number of segments in the period}$

The rationale behind this metric that can be used to evaluate the choice of Unitlength is as follows:

1. Take the example shown in Fig. 5, where Segment1 is a segment with homogeneous up trend, while Segment2 is a segment with heterogeneous up/down trends. It is quite intuitive that a segment that contains only a homogeneous trend implies the choice of UnitLength is able to capture the flow trend of a segment.
2. In other words, the higher value Increment_Stock_Ratio is, the more homogeneous the flow trend is (For example, the values of Increment_Stock_Ratio for Segment1 and Segment2 are 3.4 and 1%, respectively).
3. However, in order to prevent the UnitLength being 1 in which every segment will be homogeneous, the number of records in the segments is considered as well in order to find a larger homogenous segment. Accordingly, the reason of using the metrics $M1'$ is then explained.
4. Furthermore, since our correction models are performed for a period of time consisting of a set of segments, we have to consider all $M1'$ values in this period, resulting the use of the metric M1.

- The Sale_Ratio direction:

$M2' = |\text{Increment\_Sale\_Ratio}| * \text{Number of records in the segment}$
$M2 = \text{SUM}(M2')/\text{the number of segments in the period}$

Similar reasons as the use of M1 apply to the use of the metric M2.

With the metrics of M1 and M2, the CIMiner is able to make a choice of UnitLength with which the values of M1 and M2 are both acceptable.

5.2. Threshold T

From our method of time series pattern extraction, we know that a different threshold T would lead to different semantic representations of the data that further result in different predictions. Therefore, it poses a question—*on what grounds should a choice of T be based and said to be good or bad?*

Tracing back to the meaning of T, based on which the semantic labels, 'S'/'O'/'L', are defined. A good choice of T leads to a good quality in identifying problematic segments the CIMiner locates. However, what do we mean by the quality in identifying problematic segments?

In this paper, we employ a simple heuristic concept—'*tomorrow will be better than today*' to characterize this quality and form a formula for setting this threshold. The formula is as follows:

$$T = \frac{SUM(Diff)}{NumberofSegments}$$

Table 3
The values of UnitLength for the retailers

Retailer	Suitable UnitLength
不分區	21
T-Zone	21.28
順發	14.35
震旦	35
北區	21.42
中區	21.35
南區	28
T-Zone 天母	35.42
T-Zone 和平	28
T-Zone 忠孝	35
T-Zone 東區	21
T-Zone 湯臣	28.35
順發台中	21.35
順發高雄	28
順發新竹	14
震旦世貿	35
震旦光華	35

Table 4
The values of the threshold T for the retailers

Retailer	T value suggested by CIMiner (%)
不分區	3
T-Zone	3
順發	4
震旦	7
北	3
中	5
南	6
T-Zone 天母	5
T-Zone 和平	7
T-Zone 忠孝	4
T-Zone 東區	6
T-Zone 湯臣	4
順發台中	5
順發高雄	6
順發新竹	6
震旦世貿	12
震旦光華	3

Table 5
The results of applying the correction model

Time	27.9.1998–25.10.1998	28.10.1998–25.11.1998	1.12.1998–29.12.1998	30.12.1998–27.1.1999	31.1.1999–28.2.999
(a) Shows the change in Diff					
Prior to corrections	6.89	7.74	10.4	5.12	0.98
First revision	6.89	6.75	4.46	5.12	0.98
(b) Shows the change in labels					
Labels prior to correction	S	O	L	S	S
Labels after correction	S	S	S	S	S

Table 6
The effects of applying the correction models

	Number of 'O' corrected into 'S'	Percentage of 'O' corrected into 'S'
(a) Shows the percentage corrected for the Overstock segments		
1st correction	5	71.43
2nd correction	0	0
3rd correction	1	14.29
Total correction	6	85.72
	Number of 'L' corrected into 'S'	Percentage of 'L' corrected into 'S'
(b) Shows the percentage corrected for the Lack segments		
1st correction	6	50
2nd correction	5	41.67
3rd correction	0	0
Total correction	6	91.67

SUM(*Diff*): the summation of Diff for the segments in the periodNumberofSegment: the number of segments in the period

From this formula, for a product and a retailer, the threshold T for the next run of the CIMiner is set to be the average *Diff* of the current run of the CIMiner. In other words, with the T in the current run, the CIMiner is able to make necessary improvements on the stocks, resulting a smaller T for the next run of the CIMiner because the values of *Diff* are becoming smaller due to the improvements. However, it is still necessary to bind the values of T, ranging from 3 to 15% heuristically pre-specified, in order to prevent overfitting.

5.3. The experiment results

With the data between September, 1998 and February, 1999 we have collected for the worldwide leading printer company, we first applied the method of evaluating UnitLength as described in Section 5.1, resulting in the good choices of UnitLength for the retailers as shown in Table 3, and then with these choices of UnitLength apply the method of determining the values of the threshold T, resulting in the good choices of T for the retailers as shown in Table 4.

After determining the values of UnitLength and T for a retailer, the CIMiner is able to apply the correction models to predict the suitable amount of the stock for the retailer. For example, for a retailer 'T-Zone

和平

', with the values of UnitLength and T being 28 and 7% respectively obtained from Tables 3 and 4, Table 5(a) shows the change in *Diff* for each period and Table 5(b) shows the change in the semantic labels of the period.

Considering all of the retailers, Table 6(a) shows the percentage corrected after each correction for improving the Overstock segments, and Table 6(b) shows the percentage corrected after each correction for improving the Lack segments.

From Table 6, after three phases of corrections, the percentage corrected reached 85.72% and 91.67% for the Overstock case and Lack case respectively. In other words, our correction models exhibit a good quality in correcting unsuitable stock amounts for a product.

6. Conclusion

Traditional inventory problems concern about the amount of products a business keeps on hand to meet the demands of its customers mainly from the perspective of cash flow. The problem addressed in this paper concerns the amount of a product required at a time at a retailer for a business of a printer manufacture from a different perspective of competition encountered. That is, the information of the markets and the competitors is analyzed for obtaining the intelligence, which is unknown but good for decision making to gain competitive advantage that is very important for businesses moving forward from the Information Age to the Intelligence Age. In other words, this study aims to provide a method for devising place-prospect competitive intelligence support systems, with which a business may rationally places its products at retailers to the benefits of the business in the long run. The benefits are the followings: minimizing the cost of unnecessary holding inventory at the retailers, minimizing chances of losing possible sales, and maximizing chances of having profits.

The method, the CIMiner, presented in this paper is a combination of time series pattern extraction and time series pattern processing. For each product and retailer, time series pattern extraction aims to discover problematic segments in which the stock is not in accordance with the sale, and time series pattern processing subsequently corrects the problem and provides a prediction of the suitable amount of the product needed to be placed in the future for possessing place-prospect competitive advantage. The CIMiner also devises a systematic way of setting up the parameters (segment unit length etc.) used in the method. This CIMiner is then evaluated and exhibits a high level of quality toward achieving the goal of the problem. Our future works are to apply our method to a large amount of real data or to other CI problems and tackle the problem of inventory management from both the perspective of cash flow and the perspective of competition encountered.

Appendix A.

In this appendix, a set of screen dumps (Figs. A1–A6) is provided for giving the readers a look and feel about the execution of the CIMiner. Fig. A1 is the entry interface of the CIMiner in which the tasks of the CIMiner are invoked. In the rest of the appendix, we provide the descriptions for Figs. A2–A6:

- Fig. A2: the interface for explicitly setting the values for the variables, such as UnitLength, a retailer, the threshold T, and the percentage threshold for identifying the period that is dominated by a specific semantic label, when the users intend to overwrite the default values of these variables generated by the CIMiner.
- Figs. A3 and A4: the identification of problematic segments represented by the symbol of '??' shown in the top sub-window after the accomplishment of the analysis task. The sub-window in the middle lists the values of the parameters specific to a selected problematic segment, and the sub-window in the bottom then describes the status of this problematic segment. The button on the left-bottom corner allows the users to invoke the correction process for this segment, and the results is shown in Fig. A4.
- Fig. A5: the identification of problematic period and a list of relevant parameter values to this period. The sub-window in the bottom then describes the status of this problematic period. The button on the left-bottom corner allows the users to invoke the correction process for this period.
- Fig. A6: an integrated table that shows the semantic labels

Fig. A1. The main interface of the CIMiner.

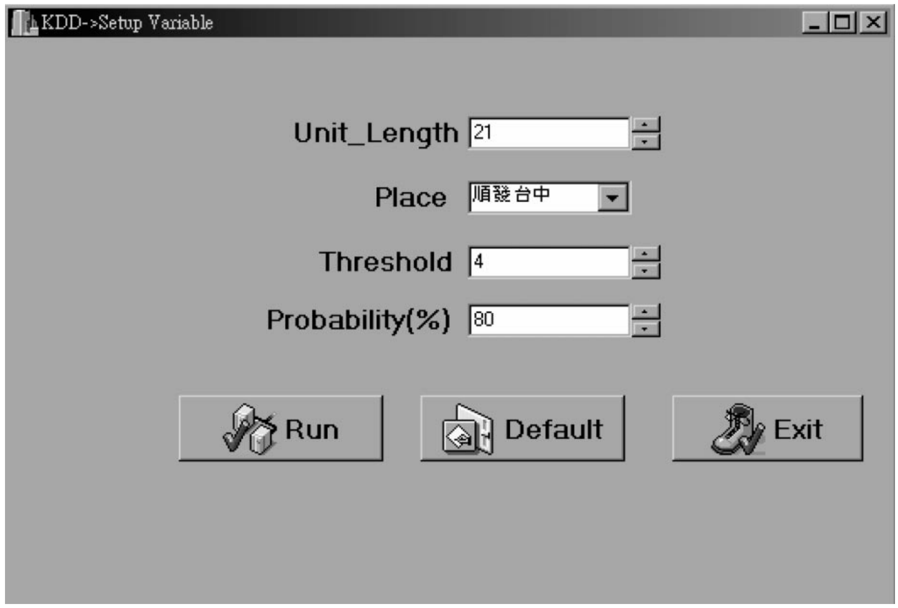

Fig. A2. The interface for explicitly setting the important system parameter.

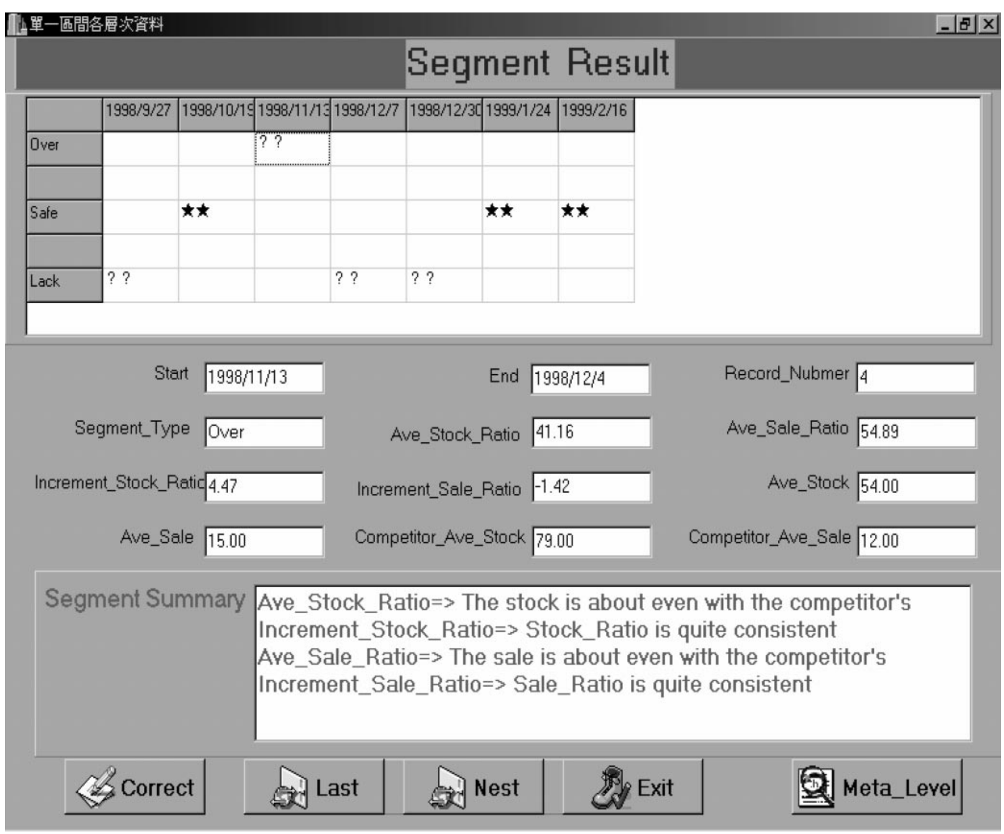

Fig. A3. The results obtained after the accomplishment of the analysis task.

Fig. A4. The results after the correction for a problematic segment.

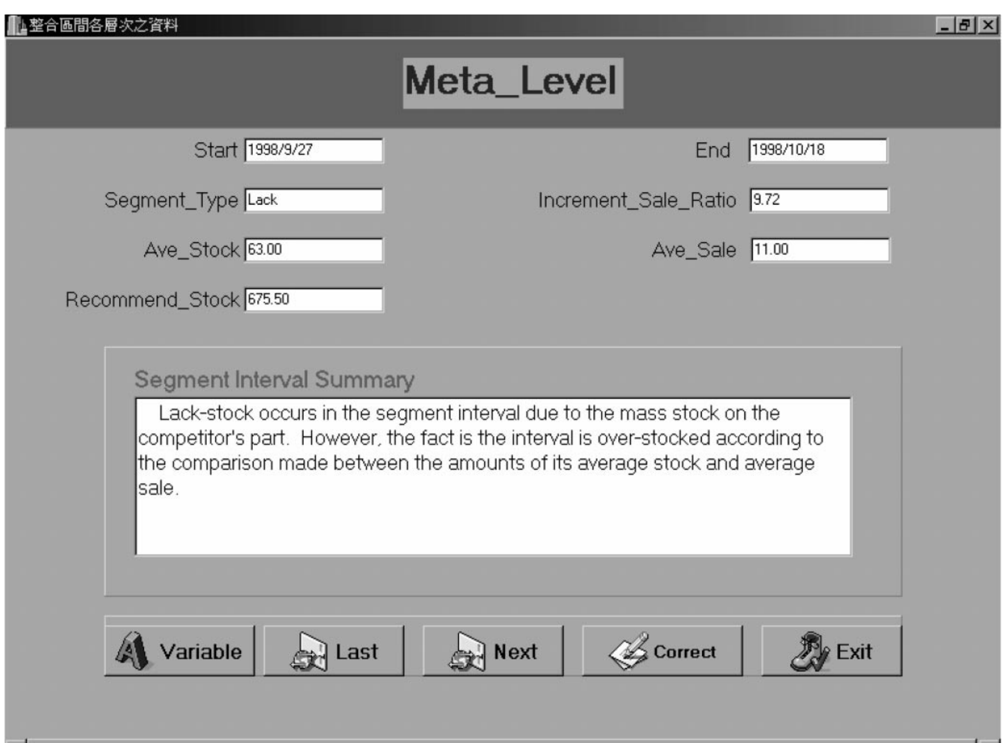

Fig. A5. The identification of the maximum period that contain a set of segments which await further correction.

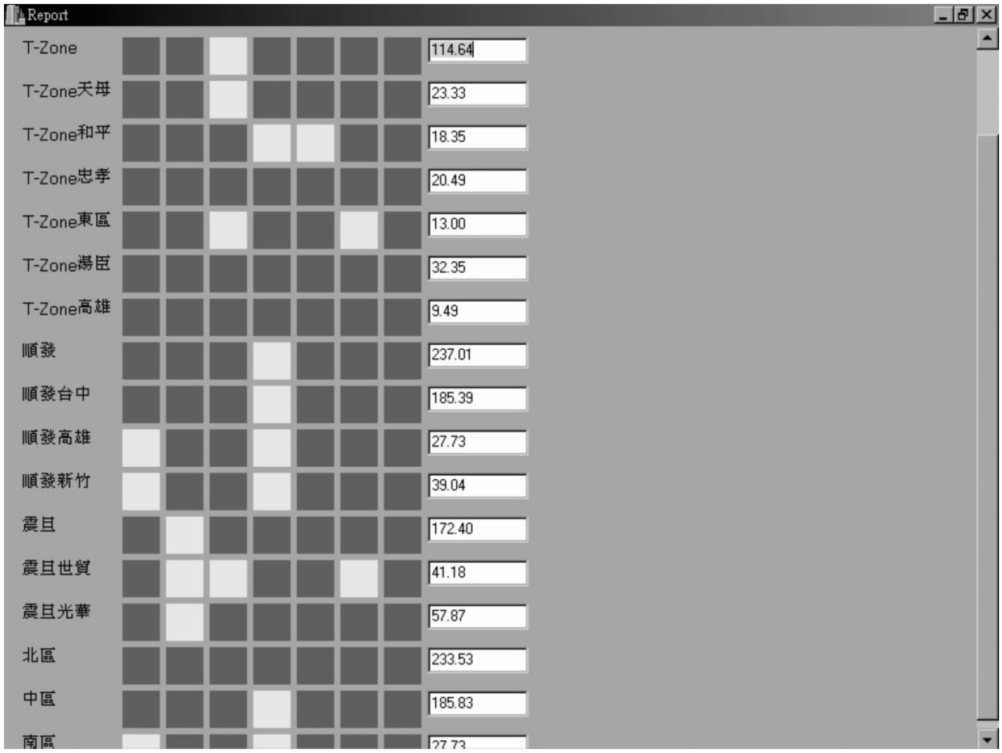

Fig. A6. An integrated table that shows the semantic labels represented by different colors for all the segments for all the retailers.

represented by different colors for all the segments for all retailers, such as green for 'S', yellow for 'L', and red for 'O'. The click of a specific retailer on the left column leads to the popup of a screen dump as shown in Fig. A3.

References

Agrawal, R., Psaila, G., Wimmers, E. L., & Zait, M. (1995). Querying Shapes of Histories, International Conference on Very Large Database.

Bollobas, B., Das, G., & Gunopulos, D. (1997). Time-Series Similarity Problems and Well-Separated Geometric Sets, Symposium on Computational Geometry.

Das, G., Gunopulos, D., & Mannila, H. (1997). Finding Similar Time Series, Pacific Rim International Conference on Knowledge Discovery and Data Mining.

Faloutsos, C., Lin, K. I., & Fastmap:, A. (1995). *Fast Algorithm for Indexing, Data Mining, and Visulization of Traditional and Multimedia Database, Proceedings of SIGMOD'*, 95.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communication of the ACM*, *39* (11), 27–34.

Han, Jiawei, Dong, G., & Yin, Y. (1999). Efficient Mining Partial Periodic Patterns in Time Series Database, International Conference on Data Engineering.

Li, C. S., Yu, Philip S., & Castelli, V. (1998). MALM: A Framework for Mining Sequence Database at Multiple Abstraction Levels, the International Conference of Information and Knowledge Management.

Qu, Yunyao, Wang, C., & Wang,X. S. (1998). Supporting Fast Search in Time Series for Movement Patterns in Multiple Scales, Seventh International Conference on Information and Knowledge Management, Washington, DC.

Schreibfeder, J. (2000). Achieving Effective Inventory Management 2nd Edition, Effective Inventory Management Inc.

Shatkay, H., & Zdonik, S. B. (1996). Approximate Queries and Representations for Large Data Sequence, International Conference on Data Engineering.

Tyson, K. W. M. (1995). Competitive knowledge development: reengineering competitive intelligence for maximum success. *Competitive Intelligence Review*, *6* (4), 14–21.