Re-mining item associations: Methodology and a case study in apparel retailing

Ayhan Demiriz [a] , Gürdal Ertek [b], Tankut Atan [c], Ufuk Kula [a]

[a] Sakarya University, Sakarya, Turkey

[b] Sabancı University, Istanbul, Turkey

[c] Işık University, Istanbul, Turkey

Re-Mining Item Associations: Methodology and a Case Study in Apparel Retailing

Ayhan Demiriz[a,*], Gürdal Ertek[b], Tankut Atan[c], Ufuk Kula[a]

[a] *Sakarya University, Sakarya, Turkey*
[b] *Sabancı University, Istanbul, Turkey*
[c] *Işık University, Istanbul, Turkey*

Abstract

Association mining is the conventional data mining technique for analyzing market basket data and it reveals the positive and negative associations between items. While being an integral part of transaction data, pricing and time information have not been integrated into market basket analysis in earlier studies. This paper proposes a new approach to mine price, time and domain related attributes through *re-mining* of association mining results. The underlying factors behind positive and negative relationships can be characterized and described through this second data mining stage. The applicability of the methodology is demonstrated through the analysis of data coming from a large apparel retail chain, and its algorithmic complexity is analyzed in comparison to the existing techniques.

Keywords: Data Mining, Association Mining, Negative Association, Apparel Retailing, Inductive Decision Trees, Retail Data

[*] Corresponding author
Email addresses: ademiriz@gmail.com (Ayhan Demiriz), ertekg@sabanciuniv.edu (Gürdal Ertek), tatan@isikun.edu.tr (Tankut Atan), ufukkula@gmail.com (Ufuk Kula)

1. Introduction

Association mining is a data mining technique which generates rules in the form of $X \Rightarrow Y$, where X and Y are two non-overlapping discrete sets. A rule is considered as significant if it is satisfied by at least a certain percentage of cases (`minimum support`) and its confidence is above a certain threshold (`minimum confidence`). Conventional association mining considers "positive" relations in the form of $X \Rightarrow Y$. However, negative associations in the form of $X \Rightarrow \neg Y$, where $\neg Y$ represents the negation (absence) of Y, can also be discovered through association mining.

Recent research has positioned association mining as one of the most popular tools in retail analytics [7]. Association mining primarily generates positive association rules that reveal complementary effects, suggesting that the purchase of an item can generate sales of other items. Yet, association mining can also be used to reveal substitution effects, where substitution means that a product is purchased instead of another one. Although positive associations have traditionally been an integral part of retail analytics, negative associations have not.

Numerous algorithms have been introduced to find positive and negative associations, following the pioneering work of Agrawal et al. [3]. Market basket analysis is considered as a motivation, and is used as a test bed for these algorithms. Price data are readily available within the market basket data and one would expect to observe its usage in various applications. Conceptually quantitative association mining (QAM) [14, 22] can handle pricing data and other attribute data. However, pricing data have not been utilized before as a quantitative attribute in quantitative association mining except in [9, 18]. Korn *et al.* [18] explore a solution with the help of singular value decomposition (SVD) and Demiriz *et al.* [9] extend the results from SVD to find item associations depending on SVD rule similarities. Quantitative association mining is not the only choice for analyzing attribute data within existing frameworks. Multidimensional association mining [14] is methodology that can be adapted in analyzing such data. The complexity of association mining increases with the usage of additional attribute data, which may include both categorical and quantitative attributes in addition to the transaction data [4]. Even worse, the attribute data might be denser compared to transaction data.

The main contribution of this paper is a practical and effective methodology that efficiently enables the incorporation of attribute data (e.g. price, category, sales timeline) in explaining positive and negative item associations, which respectively indicate the complementarity and substitution effects. To the best of our knowledge, there exists no methodological research in the data mining or information science literature that enables such a multi-faceted analysis to be executed efficiently and is proven on real world data. The core of the proposed methodology is a new secondary data mining process to discover new insights regarding positive and negative associations.

As a novel and broadly applicable concept, we introduce and define *data re-mining* as the mining of a newly formed data that is constructed upon the results of an original data mining process. The newly formed data will contain additional attributes joined with the original data mining results. In the case study presented in this paper, these attributes are related to price, item, domain and time. The methodology combines pricing as well as other information with the original association mining results through a new mining process, generating new rules to characterize, describe and explain the underlying factors behind positive and negative associations. Re-mining is fundamentally different from post-mining: post-mining only summarizes the data mining results, such as visualizing the association mining results [11]. The re-mining methodology extends and generalizes post-mining.

Our work contributes to the field of data mining in four ways:

1. We introduce a new data mining concept and its associated process, named as *Re-Mining*, which enables an elaborate analysis of both positive and negative associations for discovering the factors and explaining the reasons for such associations.
2. We enable the efficient inclusion of price data into the mining process, in addition to other attributes of the items and the application domain.
3. We illustrate that the proposed methodology is applicable to real world data, through a case study in apparel retailing industry.
4. Different ways of *re-mining*, namely exploratory, descriptive and predictive re-mining, are applied to real world data.

The re-mining framework was first introduced in [10]. This paper elaborates on the concept, introduces mathematical formalism and presents the algorithm for the methodology. The work in this paper also extends the application of predictive re-mining in addition to exploratory and descriptive re-mining, and presents a complexity analysis.

The remainder of the paper is organized as follows. In Section 2, an overview of the basic concepts in related studies is presented through a concise literature review. In Section 3, Re-Mining is motivated, defined, and framed. The methodology is put into use with apparel retail data in Section 4 and its applicability is demonstrated. In Section 5, the limitations of quantitative association mining (QAM) are illustrated with regards to the retail data used in this paper. In Section 6, the complexity of re-mining is compared to QAM, illustrating its significant advantage against QAM. Finally, Section 7 summarizes the study and discusses future directions.

2. Related Literature

One of the most common applications of association mining is literally market basket analysis (MBA), which can be used in product recommendation systems [8]. Following the notation used in [14], let $\mathcal{I} = \{I_1, I_2, ..., I_m\}$ be a set of items considered in MBA. Then each transaction (basket) T will consist of a set of items where $T \subseteq \mathcal{I}$. Each item in transaction T will have a corresponding price p, which might change from time to time i.e. p is not necessarily constant over the time span. Let X and Y be two non-overlapping sets of items, where $X \subset \mathcal{I}$ and $Y \subset \mathcal{I}$, contained in some transactions. An association rule is an implication of the following form $X \Rightarrow Y$. Assume $\mathcal{D}$ is the set of all transactions, then the support (s) of the rule is defined as the percentage of the transactions in $\mathcal{D}$ that contain both X and Y i.e. the itemset $X \cup Y$. Recall that $X \cap Y = \emptyset$. The confidence (c) of the rule $X \Rightarrow Y$ is defined as the percentage of transactions within $\mathcal{D}$ containing X, that also contain Y. In other words, confidence is the percentage of transactions containing Y, given that those transactions already contain X. Notice that this is equivalent to the definition of conditional probability.

Quantitative and multi-dimensional association mining are well-known techniques [14] that can integrate attribute data into the association mining process, where the associations among these attributes are also found. However, these techniques introduce significant additional complexity, since association mining is carried out with the complete set of attributes rather than just the market basket data. In the case of QAM, quantitative attributes are transformed into categorical attributes through discretization, transforming the problem into multi-dimensional association mining with only categorical attributes. This is an *NP-Complete* problem as shown by Angiulli *et al.* [4], meaning that the running time increases exponentially as the number of additional attributes increases linearly.

Multi-dimensional association mining works directly towards the generation of *multi-dimensional* rules. It relates all the possible categorical values of all the attributes to each other. Re-mining, on the other hand, expands *single dimensional* rules with additional attributes. In re-mining, attribute values are investigated and computed only for the positively and negatively associated item pairs, with much less computational complexity that can be solved in polynomial running time.

2.1. Negative Association Mining

In research and practice, association mining commonly refers to positive association mining. Since positive association mining has been studied extensively, only (some of) the approaches for finding negative associations are reviewed. One innovative approach [21] utilizes the domain knowledge of item hierarchy (taxonomy), and seeks negative association between items in a pairwise way. Authors in [21] propose the rule interestingness measure (RI) based on the difference between expected support and actual support: $RI = \frac{E[s(XY)]-s(XY)}{s(X)}$. A minimum threshold is specified for RI for the candidate negative itemsets, besides the minimum support threshold. Depending on the taxonomy (e.g. Figure 1(a)) and the frequent itemsets, candidate negative itemsets can be generated. For example, assuming that the itemset $\{CG\}$ is frequent in Figure 1(a), the dashed curves represent some of the candidate negative itemsets.

Tan *et al.* [24] find negative associations through indirect associations. Figure 1(b) depicts such an indirect association $\{BC\}$ via item A. In Figure 1(b) itemsets $\{AB\}$ and $\{AC\}$ are both assumed to be frequent, whereas the itemset $\{BC\}$ is not. The itemset $\{BC\}$ is said to have an *indirect association* via the item A and thus is considered as a candidate negative association. Item A in this case is called as a *mediator* for the itemset $\{BC\}$. Just like the aforementioned method in [21], indirect association mining also uses an interestingness measure -*dependency* in this case- as a threshold. Indirect mining selects as candidates the frequent itemsets that have strong dependency with their mediator.

Both methods discussed above are suitable for retail analytics and the approach in [24] is selected in this study to compute negative associations due to convenience of implementation.

2.2. Quantitative and Multi-Dimensional Association Mining

The traditional method of incorporating quantitative data into association mining is to discretize (categorize) the continuous attributes. An early work by Srikant and Agrawal [22] proposes such an approach where the continuous attributes are first partitioned and then treated just like categorical data. For this, consecutive integer values are assigned to each adjacent partition. In case the quantitative attribute has few distinct values, consecutive integer values can be assigned to these few values to conserve the ordering of the data. When there is not enough support for a partition, the adjacent partitions are merged and the mining process is rerun. [22] emphasizes rules with quantitative attributes only on the left hand side (antecedent) of the rules. However, since each partition is treated as if it were categorical, it is also possible to obtain rules with quantitative attributes on the right hand side (consequent) of the rules.

An alternative statistical approach is followed by Aumann and Lindell [6] for finding association rules with quantitative attributes. The rules found in [6] can contain statistics (mean, variance and median) of the quantitative attributes, as in our methodology.

In comparison to the aforementioned approaches, re-mining does not investigate every combination of attribute values, and is much faster than QAM. For the sake of completeness, QAM is also carried out and is compared against re-mining in Section 5.

Korn *et al.* [18] summarize the expenses made on the items through ratio rules. An example ratio rule would be "*Customers who buy bread:milk:butter spend 1:2:5 dollars on these items.*" This is a potentially useful way of utilizing the price data for unveiling the hidden relationships among the items in sales transactions. According to this approach, one can basically form a price matrix from sales transactions and analyze it via singular value decomposition (SVD) to find positive and negative associations. Ensuring the scalability of SVD in finding the ratio rules is a significant research challenge. Demiriz *et al.* [9] use ratio rule similarities based on price data to find both positive and negative item associations.

2.3. Learning Association Rules

Yao *et al.* [27] propose a framework of a learning classifier to explain the mined results. However, the described framework considers and interprets only the positive association rules and requires human intervention for labeling the generated rules as interesting or not. The framework in [27] is the closest work in the literature to the re-mining methodology proposed here. However re-mining is unique in the sense that it also includes negative associations and is suitable for the automated rule discovery to explain the originally mined results. Meanwhile, unlike in [27], the proposed approach is applied to a real world dataset as a proof of its applicability, and a through complexity analysis is carried out.

Finally, based on correlation analysis, Antoine and Zaïne [5] propose an algorithm to classify associations as positive and negative. However learning is only based on correlation data and the scope is limited to labelling the associations as positive or negative.

3. The Methodology

The proposed re-mining methodology, which transforms the post-mining step into a new data mining process, is introduced in this section. A mathematical formalism is introduced and the methodology is presented in the form of an algorithm in Figure 2. Re-mining can be considered as an additional data mining step of Knowledge Discovery in Databases (KDD) process [12] and can be conducted in explanatory, descriptive, and predictive manners. *Re-mining* process is defined as "combining the results of an original data mining process with a new additional set of data and then mining the newly formed data again".

The methodology consists of the following five steps (Figure 2):

1. Perform association mining.
2. Sort the items in the 2-itemsets.
3. Label the item associations accordingly and append them as new records.
4. Expand the records with additional attributes for re-mining.
5. Perform exploratory, descriptive, and predictive re-mining.

Conceptually, re-mining process can be extended to include many more repeating steps, since each time a new set of the attributes can be introduced and a new data mining technique can be utilized. However, in this paper, the re-mining process is limited to only one additional data mining step. In theory, the new mining step may involve any appropriate set of data mining techniques. In this paper decision tree, colored scattered plot, and several classification algorithms have been utilized.

The results of data mining may potentially be full of surprises, since it does not require any pre-assumptions (hypotheses) about the data. Making sense of such large body of results and the pressure to find surprising insights may require incorporating new attributes and subsequently executing a new data mining step, as implemented in re-mining. The goal of this re-mining process is to explain and interpret the results of the original data mining process in a different context, by generating new rules from the consolidated data. The results of re-mining need not necessarily yield an outcome in parallel with the original outcome. For example, if the original data mining yields frequent itemsets, re-mining does not necessarily yield frequent itemsets again.

The main contribution of this paper is the introduction of the re-mining process to discover new insights regarding the positive and negative associations and an extensive demonstration of its applicability. However the usage of re-mining process is not limited to gaining new insights. One can potentially employ the re-mining process to bring further insights to the results of any data mining task.

The rationale behind using the re-mining process is to exploit the domain specific knowledge in a new analysis step. One can understandably argue that such background knowledge can be integrated into the original data mining process by introducing new attributes. However, there might be certain cases that adding such information would increase the complexity of the underlying model [4] and diminish the strength of the algorithm. To be more specific, it might be necessary to find attribute associations when the item associations are also present, which requires constraint-based mining [20]. Re-mining may help with grasping the causality effects that exist in the data as well, since the input of the causality models may be an outcome of the another data mining process.

4. Case Study

In this section, the applicability of the re-mining process is demonstrated through a real world case study that involves the store level retail sales data originating from an apparel retail chain. In the case study, there was access to complete sales, stock and transshipment data belonging to a single merchandise group (men's clothes line) coming from all the stores of the retail chain (over 200 of them across the country) for the 2007 summer season. A detailed description of the retail data can be found in [10].

In this section, the application of the re-mining methodology on the retail dataset is presented. Re-mining reveals the profound effect of pricing on item associations and frequent itemsets, and provides insights that the retail company can act upon.

4.1. Conventional Association Mining

As the first step of the re-mining methodology, conventional association mining has been conducted. Apriori algorithm was run to generate the frequent itemsets with a minimum support count of 100. All the 600 items were found to be frequent. In re-mining, only the frequent 2-itemsets have been investigated and 3930 such pairs have been found. Thus frequent itemsets were used in the analysis, rather than association rules. For the illustration purposes, the top-5 frequent pairs can be found in Table 1 of [10].

The frequent pairs were then used in finding the negatively related pairs via indirect association mining. Negative relation is an indicator of product substitution. Implementing indirect association mining resulted in 5,386 negative itemsets, including the mediator items. These itemsets were reduced to a set of 2,433 unique item pairs when the mediators were removed. This indeed shows that a considerable portion of the item pairs in the dataset are negatively related via more than one mediator item.

4.2. Re-Mining the Expanded Data

Following the execution of conventional positive and negative association mining, a new data set E^* (see Figure 2) was formed from the item pairs and their additional attributes A^* for performing exploratory, descriptive and predictive re-mining. Supervised classification methods require the input data in table format, in which one of the attributes is the class label. The type of association, positive ('+') or negative ('-'), was selected as the class label in the analysis.

An item pair can generate two distinct rows for the learning set - e.g. pairs AB and BA, but this representation ultimately yields degenerate rules out of learning process. One way of representing the pair data is to order (rank) items in the pair according to a sort criterion, which is referred to as *sort_attr* in the re-mining algorithm (Figure 2). In the case study, sort attribute was selected as the price, which marks the items as higher and lower priced items, respectively.

For computing price-related statistics (means and standard deviations) a price-matrix was formed out of the transaction data $\mathcal{D}$. The price-matrix resembles the full-matrix format of the transaction data with the item's price replacing the value of **1** in the full-matrix. The price-matrix was normalized by dividing each column by its maximum value, enabling comparable statistics. The value 0 in the price-matrix means that the item is not sold in that transaction. The full price-matrix has the dimensions $2,753,260 \times 600$.

Besides price related statistics such as minimum, maximum, average and standard deviations of item prices (MinPriceH, MaxPriceH, MinPriceL, MaxPriceL, AvgPriceH_H1_L0, ..., StdDevPriceH_H1_L0, ...), attributes related with time and product hierarchy were appended, totaling to 38 additional attributes in Step 4 of the algorithm (Figure 2). Attributes related with time include first and last weeks where the items were sold (StartWeekH, EndWeekH, StartWeekL, EndWeekL) and item lifetimes (LifeTimeH, LifeTimeL). Attributes related with product hierarchy include the categories and the merchandise subgroups of the items (CategoryH, CategoryL, MerchSubGrpH, MerchSubGrpL). Together with the record ID, the two item IDs and the type of association, re-mining was carried out with a total of 42 attributes with 6,363 records. All the additional attributes have been computed for all sorted item-pairs through executing relational queries.

Once the new dataset E^* is available for the re-mining step, any decision tree algorithm can be run to generate the descriptive rules. Decision tree methods such as C5.0, CHAID, CART are readily available within data mining software in interactive and automated modes.

If decision tree analysis is conducted with all the attributes, support related attributes would appear as the most significant ones, and data would be perfectly classified. Therefore it is necessary to exclude the item support attributes from the analysis.

One example of the rule describing the '+' class is Rule 1 in Figure 3, which can be written as follows:

IF LifeTimeL $<$ 22 AND LifeTimeH $\geq$ 28 THEN '+'.

This rule reveals that if the lifetime of the lower-priced item is less than 22 weeks, and the lifetime of the higher-priced item is greater than or equal to 28 weeks, then there is a very high probability of this item pair to exhibit '+' association (having the target class '+').

An example of '-' class rule is Rule 2 in Figure 3:

IF LifeTimeL $<$ 22 AND LifeTimeH $<$ 28 AND CorrNormPrice_HL $<$ 0.003 THEN '-'.

This rule reveals that if the lifetime of the lower-priced item is less than 22 weeks, and the lifetime of the higher-priced item is less than 28 weeks, the correlation between the prices of these items is less than 0.003 (practically translating into negative correlation), then there is a very high probability of this item pair to exhibit '-' association (having the target class '-').

Ceteris paribus, given everything else being the same, retail managers would prefer positive associations between items. This translates into traversing the darker nodes in Figure 3, where nodes are darkened with increasing probability of positive associations. In the visual mining of the decision tree, one is interested in the node transitions where a significant change takes place in the node color. So in Figure 3, a transition from a light-colored node to a dark node is interesting. By observing Figure 3, several policies have been developed and are presented below. The policies listed below are named based on the level of the tree that they are derived. They are also marked on the tree in Figure 3.

Policy 1: "LifeTimeL $\geq$ 22". The lower-priced item should have a life time greater than or equal to 22 weeks.

Policy 2: "LifeTimeL $\geq$ 28". It's even better that the low priced item has life time greater than or equal to 28 weeks, rather than 22 weeks. This would increase its probability of having positive associations with higher priced items from 0.73 to 0.92 (shown with darker node).

Policy 3: "IF LifeTimeL $<$ 22 THEN LifeTimeH $\geq$ 28". If it is not possible to apply Policy 1, one should make sure to have higher priced items have a life time greater than or equal to 28 weeks. This significantly increases the probability of having positive associations from 0.50 to 0.86.

Policy 4: "IF LifeTimeL $\geq$ 22 AND LifeTimeL $<$ 28 THEN CorrNormPrice_HL $\geq$ 0.003". If it is possible to apply Policy 1, but not Policy 2, then the pricing policy of synchronized price decrease can still increase the probability of positive associations. Policy 4 suggests that the normalized prices should have positive correlation, so that positive associations will emerge. Since the prices are monotonically non-increasing through the season, this insight suggests that the prices of high-price and low-price items should be decreased together (so that there will be positive correlation between the normalized prices).

Policy 5: "IF LifeTimeL $<$ 22 AND LifeTimeH $<$ 28 THEN CorrNormPrice_HL $\geq$ 0.026". If it is possible to apply Policy 1, but not Policy 3, then one should again apply Policy 4, but marking down the prices of the items such that a higher correlation (at least 0.026, rather than 0.003) exists between the prices.

Policy 6: When it is possible to apply Policy 1, but not Policy 2 or Policy 4, there is still good news. For low-priced items in Category "C006", a life time between 22 and 28 weeks yields a fairly high (0.77) probability of positive association (more than double the probability for items in other categories that have the same life times). So, applying Policy 1 just for

Category "C006", making sure that the low-priced item has a life cycle of at least 22 weeks, guarantees a 0.77 probability of positive association with at least one other higher priced item.

The policies obtained through visual analysis of decision tree (Figure 3) should be handled with caution: The policies suggested above should be put into computational models for numerically justifying their feasibility. For example, Policy 1 suggests that the lower-priced item should have a lifetime of at least 22 weeks. But is such a long lifetime financially justified, especially when there are many lower-priced items? If the additional costs that will arise are not high enough to offset the additional generated sales, then Policy 1 may be justified; but this is not guaranteed a priori, without a formal fact-based numerical analysis. Thus, the policies should not be applied in isolation, but taking into consideration their interactions.

4.3. Exploratory Re-Mining

Exploratory re-mining is also conducted by creating potentially informative scatter plots. As seen from Figure 4, positive ("TargetClass=+") and negative ("TargetClass=-") associations are well-separated in different regions of the plot. For example, if StartWeekL is around the fifth week of the season then the associations are mainly positive. In addition, a body of positive associations up to 'StartWeekL=15' can be seen in Figure 4. It means that there is a high possibility of having a positive association between two items when the lower priced item is introduced early in the season. Since basic items are usually introduced early in the season, higher chance of positive association between two items can be hypothesized when the lower priced item is a basic one vs. a fashion one.

Visual analysis of Figure 5 yields new policies. In the figure, the dark blue points indicate positive associations, and light grey points indicate negative associations. It is easy to see the very heavy concentration of blue dots in the region where the initial (maximum) item price is between 14 and 18 TL (Turkish Liras) ($14 < \text{MaxPriceL} < 18$). The concentration of grey dots in that region is very dense, suggesting Policy 5. Also, when scanning the y axis (MaxPriceL), one can observe that items with an initial price tag greater than 70 TL almost always have positive associations with lower-priced items in the price range of 38 to 42 TL, suggesting Policy 6.

Policy 7: Keep the initial price of low-priced items in the range of 14 to 18 TL. This will enable them to generate new sales, thanks to their positive associations with many higher priced items.

Policy 8: Position the items with price tag between 38 to 42 TL together with (related) items that have an initial price tag over 70 TL.

It is clear that the policies generated by the visual analysis of the decision tree and a sample colored scatter plot complement each other. This suggests that one should carry out a variety of analyses in the re-mining stage.

4.4. Predictive Re-Mining

In predictive re-mining, the aim is to predict (based on their attributes) whether two items have positive or negative associations. To show the applicability of re-mining, various predictive methods in SAS Enterprise Miner are used in this study. Support Vector Machines (SVM) [23], Memory Based Reasoning (MBR), Neural Networks (NN) [26], Gradient Boosting [13], AutoNeural and Decision Tree [28] nodes are utilized in predictive analysis. The data set is partitioned to training set (50%), validation set (20%) and test set (30%) before predictive methods.

The accuracy results are reported in Table 1. NN model is the clear winner among the predictive models. Accuracy rates of the test set from four different models over 0.70 are an indication of accurate item association predictions based on the attribute data. Both positive and negative associations can be predicted accurately by utilizing attribute data, suggesting that re-mining approach can be successfully used for predictive purposes.

In addition to accuracy rates, receiver operating characteristic (ROC) curves (see [28]) are given in Figure 6. Sensitivity and specificity are defined as the true positive rate and the true negative rate, respectively. By definition, better classifiers have ROC curves towards the upper left of the plots. Therefore NN model has the best ROC curve and the decision tree model has the second best ROC curve among the predictive models. Similarly the cumulative lift plots [17] are depicted in Figure 7. The most upper right plot represents the best classifier in terms of lift. NN is again the best classifier and SVM performs as the next best model.

Variable importance is another statistic regarding classification models. Table 2 reports importance statistics of training and validation sets separately from the decision tree model. In light of Table 2 one can conclude that the correlation-based attributes play important roles along with the price-related attributes. Nevertheless, item attributes have predictive properties in terms of item associations. The attribute data can and should be used in determining the item associations. The plain transaction data might be less informative in terms of item associations.

The evidence presented in this section suggests that the use of classification models is justified and Policy 9 presents an algorithm for incorporating classification models into retail markdown planning for the next season:

Policy 9: For every item, compute the values of the important variables in Table 2. Then put these into the classification model and predict the sign of association between every item pair. For items that are predicted to have positive associations, create clusters based on merchandise group, start time, and initial price. Present the clusters of items together in marketing campaigns and in stores.

5. Comparison with Quantitative Association Mining

As mentioned earlier, as an alternative to re-mining, additional attribute data used can also be incorporated through QAM. This alternative analysis has also been conducted to illustrate its limitations on analyzing retail data. Quantitative association mining has been studied extensively in the literature and some of its major applications, such as [22], were reviewed in Section 2.2. In the case study, an initial analysis of the price data suggested that there are few price levels for each of the products. Therefore a straight-forward discretization, which does not require a complex transformation, was possible. One of the most important steps in the QAM is the discretization step. Instead of utilizing a more complex scheme for QAM, in the re-mining case study, the item and price information were conjoined into a new entity.

The two main seasons in apparel retailing, winter and summer, have approximately equal length. As a common business rule, prices are not marked down too often. Usually, at least two weeks pass by between subsequent markdowns. Thus, there exist a countable set of price levels within each season. There might be temporary sales promotions during the season, but marked down prices remain the same until the next price markdown. Prices are set at the highest level in the beginning of each season, falling down by each markdown. If the price data of a product is normalized by dividing by the highest price, normalized prices will be less than or equal to one. When two significant digits are used after the decimal point, prices can be easily discretized. For example, after an initial markdown of 10%, the normalized price will be 0.90. Markdowns are usually computed on the original highest price. In other words, if the second markdown is 30% then the normalized price is computed as 0.70.

After using this discretization scheme, 3,851 unique product-normalized price pairs have been obtained for the 600 unique products of the original transaction data. Each product has six price levels on the average. The highest number of price levels for a product is 14 and the lowest number of price levels is two. This shows that markdowns were applied to all the products and no product has been sold at its original price throughout the whole season. Technically, a discretized price can be appended to a corresponding product name to create a new unique entity for the QAM. For example appending 0.90 to the product name 'A' after an underscore will create the new entity 'A_0.90'. One can easily utilize the conventional association mining to conduct a discretized QAM after this data transformation.

The top-10 frequent pairs and their support counts can be found in Table 2 of [10]. As can be observed from that table, a large portion of the frequent purchases occurs at the discounted prices. Among the top-10 frequent item pairs, only the sixth one has full prices for both of the items. The remaining pairs are purchased at marked down prices.

The retail company does not allow price differentiations by locations. In other words, an item will have the same price across all the stores. Quantitative association mining can identify negative associations between items for value combinations that actually never occur. For example, even though an item A is sold only at the normalized price of 0.70 in the time interval that B is sold, QAM can still suggest other price combinations of A and B, such as (A_1.00, B_1.00) as negative item pairs. Thus, many item-price combinations will have negative associations due to the nature of the business and the way QAM operates, yielding misleading results.

Even though both positive and negative QAM can be run on any given dataset conceptually, it is not guaranteed to yield useful outcomes. Alternatively, re-mining operates only on the confirmed positive and negative quantitative associations and does not exhibit the discussed problem.

6. Complexity Analysis

In this section, the complexity of the re-mining algorithm is analyzed in comparison to the conventional QAM. First, the complexity of re-mining is discussed, based on results from the literature on the Apriori algorithm. Then, the complexity of QAM is discussed, based on two milestone papers [4, 25], and results from the computational complexity literature.

Before proceeding with the complexity analysis, it should be remarked that QAM computes detailed itemsets/rules, that contain specific information that contain *atomic conditions*, such as "Attribute A_1 taking the value of u_{1j}, and attribute A_2 taking the value of u_{2j}". An atomic condition is in the form $A_i = u_{ij}$ for attribute A_i, for categorical attributes, and in the form $A_i \in [l, u]$ for numeric attributes. A frequent itemset in the context of QAM is as a *condition* C on a set of distinct attributes $A_1, \ldots, A_N$, where N is the total number of attributes. C is in the form $C = C_1 \wedge C_2 \ldots \wedge C_N$, where is C_i is an atomic condition on A_i, for each $i = 1, \ldots N$ [4]. In [25] the term *pattern* is used instead of condition. An association rule in the context of QAM is an expression in the form *Antecedent* $\Rightarrow$ *Consequent*, where both *Antecedent* and *Consequent* are conditions. However, this type of an output still requires aggregating the results to obtain the summary statistics that re-mining provides. Alternatively, re-mining starts with only the interesting rules (frequent itemsets in the case study) obtained through Apriori, and then enriches these results with statistics computed for the itemsets through database queries. So re-mining eliminates the computation of specific detailed rules, and computes only the interesting itemsets and their statistics.

6.1. Complexity of Re-mining

The problem of finding the frequent itemsets (and the related association rules), itemsets that appear in at least s percent of the transactions, is the most fundamental analysis in association mining. The standard algorithm for this problem is the Apriori algorithm, first proposed in [3], and then improved in numerous succeeding studies, including [2, 15].

A new line of research approaches the frequent itemset discovery problem from a graph-theoretic perspective, achieving even further efficiency in running times. [19] extends earlier work in [16], posing the problem as a frequent subgraph discovery problem. The goal is finding all connected subgraphs that appear frequently in a large graph database. The authors report that their algorithm scales linearly with respect to the size of the data set.

Even though new algorithms are continuously being developed with efficiency increases, the reference complexity result regarding the Apriori algorithm (considering its numerous variants) is that it has an upper bound of $O(|C| \cdot |\mathcal{D}|)$ and a lower bound of $O(k \cdot log(|\mathcal{I}|/k))$ on running time [1]. In these expressions, $|C|$ denotes the sum of sizes of candidates considered, $|\mathcal{D}|$ denotes the size of the transactions database $\mathcal{D}$, $|\mathcal{I}|$ is the number of items, and k is the size of the largest set.

Now the complexity of re-mining will be analyzed following a fixed-schema complexity perspective [25], where the set of attributes A is fixed (with size N) and the complexity is derived relative to the number of tuples in the set of items $\mathcal{I}$. In the presentation of the re-mining methodology, only the itemsets of size 2 were selected. Thus the total number of frequent itemset candidates is $|\mathcal{I}|(|\mathcal{I}|-1)/2$, and its order is $O(|\mathcal{I}|^2)$. Hence $|C|$ is also by $O(|\mathcal{I}|^2)$, and the upper bound for the running of the Apriori algorithm is $O(|\mathcal{I}|^2 \cdot |\mathcal{D}|)$. In the re-mining phase, the data is enriched with statistics for the two-itemsets. There are N attributes for which statistics will be computed. For each of these attributes, there will be a query that filters the transactions with the itemset of interest, which will take $O(|\mathcal{D}|)$. Then there will be the computation of the statistic, which will require a single pass over the selected transactions for computing the mean, and two passes for computing the standard deviation. So the upper bound for the computation of each statistic is $O(|\mathcal{D}|^2)$, and this overrides the time for the filter query. Since there are N attributes, the running time for the computation of the statistics is $O(N \cdot |\mathcal{D}|^2)$. When this is combined with the running time of the Apriori, the upper bound for the whole re-mining process with $k = 2$ is $O(|\mathcal{I}|^2 \cdot |\mathcal{D}| + N \cdot |\mathcal{D}|^2)$.

6.2. Complexity of Quantitative Association Mining

The alternative to re-mining is carrying out quantitative association mining (QAM), and aggregating the results to obtain statistics for the itemsets. The two most comprehensive studies in the literature regarding the complexity of QAM are [4, 25].

The most fundamental result in both papers is that QAM is *NP-complete*, even in databases without nulls. The proof of this theorem is through reducing the CLIQUE problem to the QAM problem. The decision version of the CLIQUE problem tests whether a given graph contains a k-clique, i.e., a complete subgraph with size k, that has all its elements connected.

Now the complexity of QAM is analyzed following a fixed-schema complexity perspective, just as was done for re-mining. The additional effort to compute the summary statistics from the quantitative association rules will also be considered.

According to Wijsen and Meersman, the fixed-schema complexity of QAM is $O(|\mathcal{D}|^{(2(N+1)+1))}) = O(|\mathcal{D}|^{(2N+3)}) = O(|\mathcal{D}|^{2N})$. The number of attributes has been taken as $N+1$, rather than N, to include a new variable for the item label; but this does not change the complexity much. Once all the rules are generated, the time to filter for two-itemsets and compute the statistics is the same as in re-mining, namely $O(N \cdot |\mathcal{D}|^2)$. So the complexity of the whole process is $O(|\mathcal{D}|^{2N} + N \cdot |\mathcal{D}|^2)$. Recall that the complexity of re-mining for $k = 2$ is $O(|\mathcal{I}|^2 \cdot |\mathcal{D}| + N \cdot |\mathcal{D}|^2)$.

The running time of QAM increases exponentially with N. This suggests that re-mining becomes much more efficient as N increases. Also, the running time of QAM takes power of $|\mathcal{D}|$, which suggests that re-mining becomes much more efficient for larger databases.

7. Conclusion

A novel methodology, namely re-mining, has been introduced in this study to enrich the original data mining process. A new set of data is added to the results of the traditional association mining and an additional data mining step is conducted. The goal is to describe and explore the factors behind positive and negative association mining and to predict the type of associations based on attribute data. It is shown that not only categorical attributes (e.g. category of the product), but also quantitative attributes such as price, lifetime of the products in weeks and some derived statistics, can be included in the study while avoiding *NP-completeness*.

The re-mining methodology has been demonstrated through a case study in apparel retail industry and its practicality has been proven for the problem at hand. Descriptive, exploratory and predictive re-mining can be implemented to reveal interesting patterns previously hidden in data. The case study revealed interesting outcomes such as "negative associations are usually seen between fashion items" and "the price of an item is an important factor for the item associations in apparel retailing."

As a future study, re-mining can be applied for different merchandise groups and for successive seasons, testing its robustness w.r.t. sampling bias and temporal change, respectively. Data coming from other retailers, with diverse characteristics, can be used for further experimental evidence. Finally, data coming from different domains can be used for conforming the applicability of the re-mining process in those domains.

Acknowledgement

This work is financially supported by the Turkish Scientific Research Council under Grant TUBITAK 107M257.

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In U. Fayyad and et al, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, Menlo Park, CA, 1996.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference,Santiago, Chile*, pages 487–499, 1994.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *SIGMOD Conference*, pages 207–216. ACM Press, 1993.

[4] Fabrizio Angiulli, Giovambattista Ianni, and Luigi Palopoli. On the complexity of inducing categorical and quantitative association rules. *Theoretical Computer Science*, 314(1-2):217–249, February 2004.

[5] Maria-Luiza Antonie and Osmar R. Zaïane. An associative classifier based on positive and negative rules. In Gautam Das, Bing Liu, and Philip S. Yu, editors, *DMKD*, pages 64–69. ACM, 2004.

[6] Yonatan Aumann and Yehuda Lindell. A statistical theory for quantitative association rules. *J. Intell. Inf. Syst.*, 20(3):255–283, 2003.

[7] Tom Brijs, Gilbert Swinnen, Koen Vanhoof, and Geert Wets. Building an association rules framework to improve product assortment decisions. *Data Min. Knowl. Discov.*, 8(1):7–23, 2004.

[8] Ayhan Demiriz. Enhancing product recommender systems on sparse binary data. *Journal of Data Mining and Knowledge Discovery*, 9(2):147–170, September 2004.

[9] Ayhan Demiriz, Ahmet Cihan, and Ufuk Kula. Analyzing price data to determine positive and negative product associations. In Chi Leung,

Minho Lee, and Jonathan Chan, editors, *Neural Information Processing*, volume 5863 of *LNCS*, pages 846–855. Springer, 2009.

[10] Ayhan Demiriz, Gurdal Ertek, Tankut Atan, and Ufuk Kula. Re-mining positive and negative association mining results. In Petra Perner, editor, *Advances in Data Mining. Applications and Theoretical Aspects*, volume 6171 of *LNCS*, pages 101–114. Springer, 2010.

[11] Gürdal Ertek and Ayhan Demiriz. A framework for visualizing association mining results. In Albert Levi, Erkay Savas, Hüsnü Yenigün, Selim Balcisoy, and Yücel Saygin, editors, *ISCIS*, volume 4263 of *Lecture Notes in Computer Science*, pages 593–602. Springer, 2006.

[12] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34, Menlo Park, CA, 1996. AAAI Press.

[13] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367 – 378, 2002.

[14] Jiawei Han and Micheline Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2nd edition, 2006.

[15] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, *SIGMOD Conference*, pages 1–12. ACM, 2000.

[16] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data.

In Djamel A. Zighed, Henryk Jan Komorowski, and Jan M. Zytkow, editors, *PKDD*, volume 1910 of *Lecture Notes in Computer Science*, pages 13–23. Springer, 2000.

[17] YongSeog Kim and W. Nick Street. An intelligent system for customer targeting: a data mining approach. *Decision Support Systems*, 37(2):215 – 228, 2004.

[18] Flip Korn, Alexandros Labrinidis, Yannis Kotidis, and Christos Faloutsos. Quantifiable data mining using ratio rules. *VLDB J.*, 8(3-4):254–266, 2000.

[19] Michihiro Kuramochi and George Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.*, 16(9):1038–1051, 2004.

[20] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 13–24, New York, NY, USA, 1998. ACM.

[21] A. Savasere, E. Omiecinski, and S.B. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the 14th International Conference on Data Engineering*, pages 494–502, 1998.

[22] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and Inder-

pal Singh Mumick, editors, *SIGMOD Conference*, pages 1–12. ACM Press, 1996.

[23] Aixin Sun, Ee-Peng Lim, and Ying Liu. On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191 – 201, 2009. Information product markets.

[24] Pang-Ning Tan, Vipin Kumar, and Harumi Kuno. Using sas for mining indirect associations in data. In *Western Users of SAS Software Conference*, 2001.

[25] Jef Wijsen and Robert Meersman. On the complexity of mining quantitative association rules. *Data Min. Knowl. Discov.*, 2(3):263–281, 1998.

[26] Bo K. Wong, Thomas A. Bodnovich, and Yakup Selvi. Neural network applications in business: A review and analysis of the literature (1988-1995). *Decision Support Systems*, 19(4):301 – 320, 1997.

[27] Yiyu Yao, Yan Zhao, and R. Brien Maguire. Explanation-oriented association mining using a combination of unsupervised and supervised learning algorithms. In Yang Xiang and Brahim Chaib-draa, editors, *Canadian Conference on AI*, volume 2671 of *Lecture Notes in Computer Science*, pages 527–531. Springer, 2003.

[28] Huimin Zhao. A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, 43(3):809 – 826, 2007. Integrated Decision Support.

Table 1: Classification Results

Model Node	Test Set Accuracy Rate	Training Set Accuracy Rate	Validation Set Accuracy Rate
Neural Networks	0.82	0.86	0.83
Decision Tree	0.79	0.84	0.85
SVM	0.76	0.79	0.78
MBR	0.70	0.77	0.72
Gradient Boosting	0.62	0.62	0.62
AutoNeural	0.59	0.61	0.60

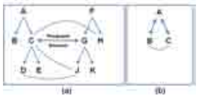

Figure 1: (a) Taxonomy of Items and Associations [21]; (b) Indirect Association

Table 2: Decision Tree Model: Variable Importance

Variable Name	Number of Rules	Importance	Validation Importance
CorrNormPrice_HL	5	1.00	1.00
LifeTimeH	3	0.90	0.82
CategoryL	2	0.81	0.78
StdDevPriceL_H0_L1	1	0.48	0.45
CorrWeeklySales_HL	2	0.46	0.52
LifeTimeL	2	0.45	0.31
StdDevPriceH_H0_L1	1	0.39	0.23
CategoryH	2	0.37	0.24
EndWeekL	2	0.33	0.32
MerchSubGrpH	1	0.23	0.18
MaxPriceH	1	0.21	0.00
AvgPriceL_H1_L0	1	0.18	0.19

Figure 2: Re-Mining Algorithm

Inputs

I:	set of items; $i \in I$
$\mathcal{D}$:	set of transactions, containing only the itemset information
min_sup:	minimum support required for frequent itemsets
min_items:	minimum number of items in the itemsets
max_items:	maximum number of items in the itemsets
$sort\_attr$:	the attribute used to sort the items within the 2-itemsets, e.g. price of an item in retailing

Definitions

L_k^+:	set of frequent k-itemsets that have positive association; $l \in L_k^+$
L_k^-:	set of k-itemsets that have negative association; $l \in L_k^-$
L_k^*:	set of all k-itemsets from association mining; $l \in L_k^*$
$l(j)$:	jth element in itemset l; $l \in L_k^*$
$i.attr$:	value of attribute $attr$ for item i; $i \in I$
E^+:	set of records for re-mining, that contain *positive* associations; $r \in E^+$
E^-:	set of records for re-mining, that contain *negative* associations; $r \in E^-$
E^*:	set of all records for re-mining, $r = (l(1), l(2), \Gamma), r \in E$, $\Gamma \in \{+, -\}$
A_n:	additional attribute n introduced for re-mining; $n = 1 \dots N$
A^*:	set of all attributes introduced for re-mining; $A^* = \cup A_n$

Functions

apriori($\mathcal{D}, min\_items, max\_items, min\_sup$):

apriori algorithm that operates on $\mathcal{D}$ and generates frequent itemsets with minimum of min_items items, maximum of max_items, and a minimum support value of min_sup

indirect_assoc($\mathcal{D}, min\_items, max\_items$):

indirect association mining algorithm that operates on $\mathcal{D}$ and generates itemsets that have *negative* association, with minimum of min_items items and maximum of max_items

$\mathbf{f}_{A_n}(l)$:

function that computes the value of attribute A_n for a given itemset l, $l \in L_k^*$

swap(a, b) :

function that swaps the values of a and b

Algorithm: Re-mining

```
1. Perform association mining.
        $L_2^+ = \mathbf{apriori}(\mathcal{D}, 2, 2, min\_sup)$
        $L_2^- = \mathbf{indirect\_assoc}(\mathcal{D}, 2, 2)$
2. Sort the items in the 2-itemsets.
        for all $l \in L_2^+, L_2^-$
            if $l(1).sort\_attr < l(2).sort\_attr$ then
                $l = (l(1), l(2)) = \mathbf{swap}(l(1), l(2))$
3. Label the item associations accordingly and append them as new records.
        $E^+ = \{r : r = (l(1), l(2), +), \forall l \in L_2^+\}$
        $E^- = \{r : r = (l(1), l(2), -), \forall l \in L_2^-\}$
        $E^* = E^+ \cup E^-$
4. Expand the records with additional attributes for re-mining.
        for $n = 1 \dots N$
            for all $r \in E^*$
                $r = (r, \mathbf{f}_{A_n}(l))$
5. Perform exploratory, descriptive, and predictive re-mining.
```

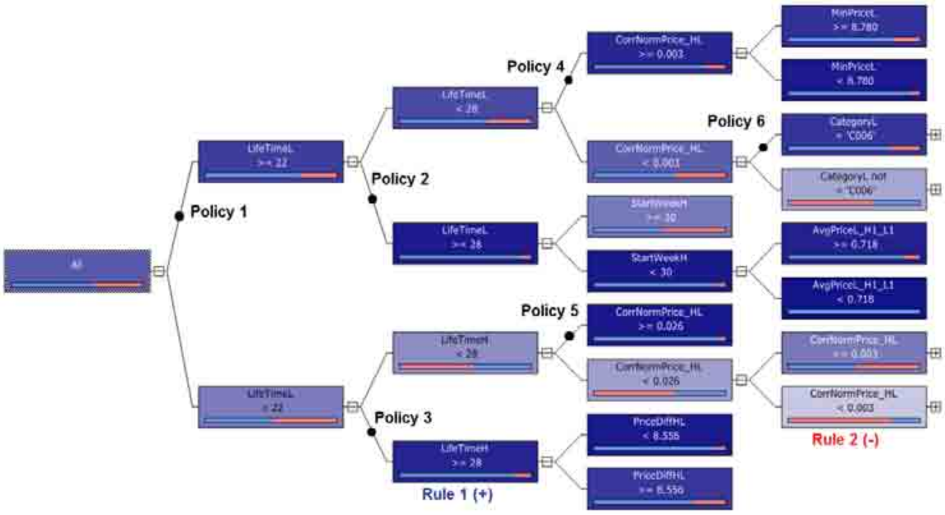

Figure 3: An Illustrative Decision Tree Model in Re-Mining

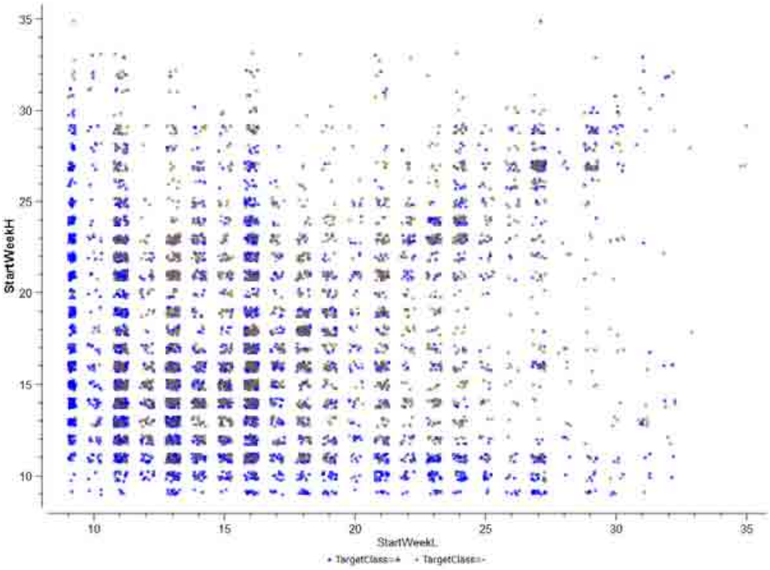

Figure 4: Exploratory Re-Mining Example: Analyzing Item Introduction in Season

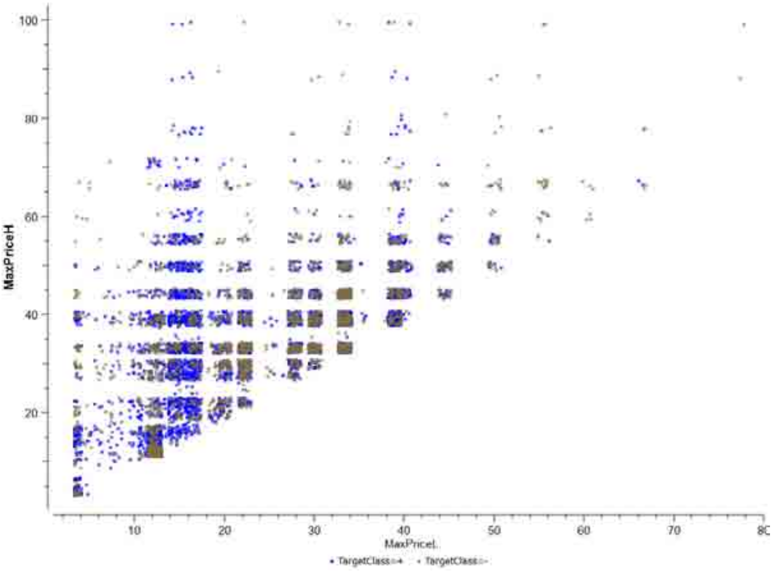

Figure 5: Exploratory Re-Mining Example: Effect of the Maximum Item Price

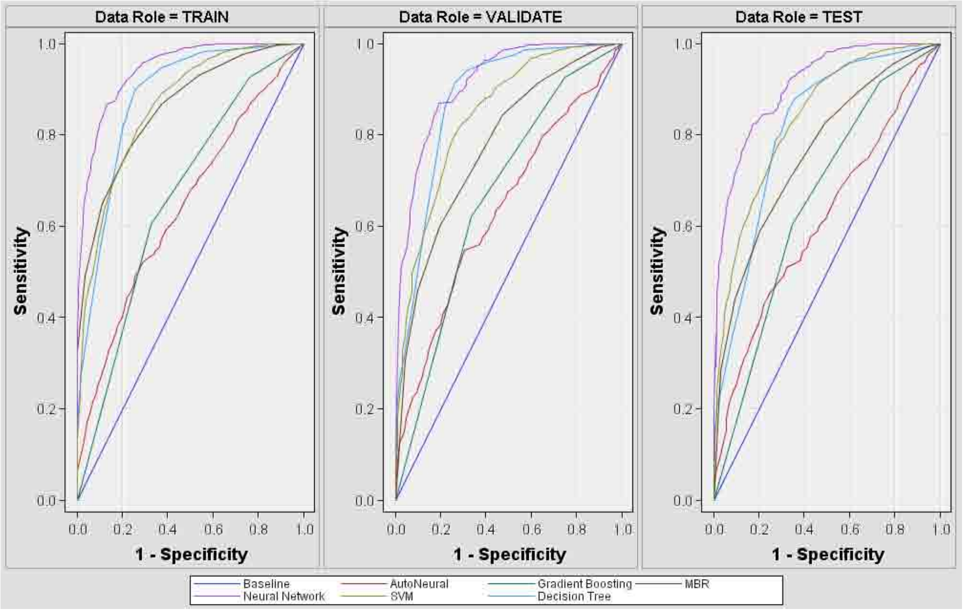

Figure 6: ROC Curves

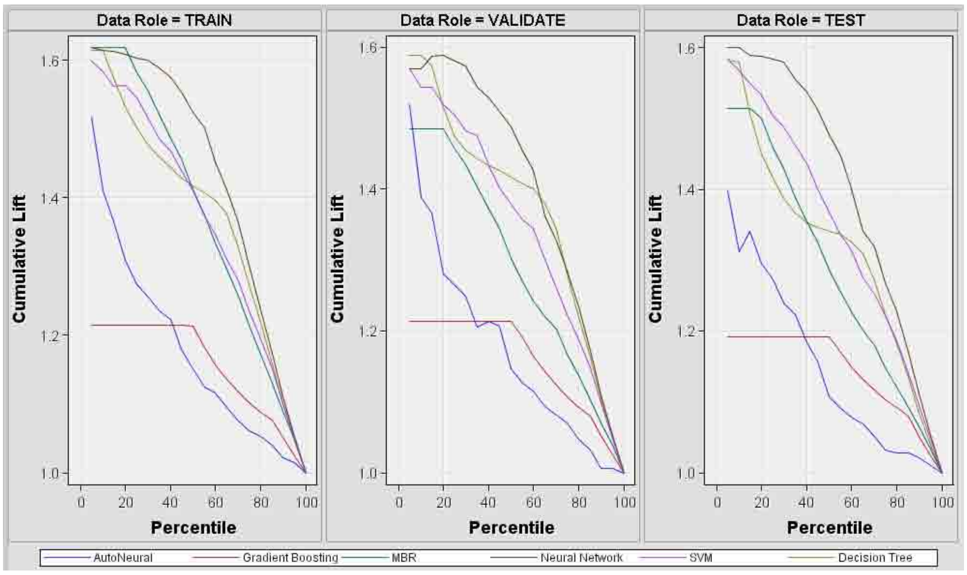

Figure 7: Cumulative Lift Plots