# CLUSTERING PRODUCTS UNDER PAIRWISE POSITIVE AND NEGATIVE ASSOCIATION CONSTRAINTS IN RETAILING

| Ayhan Demiriz | Betül Ekizoğlu | Ufuk Kula |
| --- | --- | --- |
| Dept. of Industrial Eng., | Dept. of Industrial Eng., | Dept. of Industrial Eng., |
| Sakarya Uni., Turkey | Sakarya Uni., Turkey | Sakarya Uni., Turkey |
| ademiriz@gmail.com | ekizoglubetul@gmail.com | ufukkula@gmail.com |

**Abstract**

Finding the item associations and corresponding rules are the main purposes of association mining algorithms. The resulting rules of positive associations can easily be applied for organizing supermarket shelves. However negative associations are rarely used in application domains due to the lack of universally accepted methods for finding concrete negative associations. Clustering is a proven method in data analysis to group similar objects while separating dissimilar ones. The idea of using both positive and negative item associations in a constrained clustering framework is proposed and implemented in this paper. The products of an apparel retailing firm are clustered based on their weekly sales figures in addition to the constraints originating from pairwise positive and negative product associations. A minimum cluster size constraint is also introduced for ensuring the practicality of the clustering results in terms of generating multi-product groups to run markdown optimization algorithms on such product groups in the later stages of our overall revenue management research project in which forecasting of product sales can be conducted within each cluster in a multivariate way.

## 1 Introduction

Markdown optimization is an essential tool for increasing the revenue of an apparel retailer (MR01). Single item or two-item markdown optimizations are common in application (BGc09). There are computationally challenging issues besides determining the related group of products in a multi-product markdown optimization problem. It is thus essential to form right product clusters to apply a successful multi-product markdown optimization. When the clusters are formed one of the objectives should be to generate a grouping that products within each group will not adversely affect each other in terms of markdown decisions. Product hierarchy can naturally be used in determining such multi-product groups. For example, products belong to a category can form such a group. Since the product hierarchy is known a priori, this type of grouping does not yield any new information in terms of product associations. Using only the product hierarchy is rather limiting possible combinations to create multi-product groups.

Finding item associations is the main task of well-known association mining algorithms. Association mining is a data mining process in which the goal is to find rules in the form of $X \Rightarrow Y$, where $X$ and $Y$ are two non-overlapping sets of items or events, depending on the domain. The significance of the rule comes from the fact that it is satisfied by at least a percentage of cases specified beforehand (`minimum support`) and the confidence of the rule is above a certain threshold (`minimum confidence`). Conventional association mining considers “positive” relations as in the rule $X \Rightarrow Y$. Such rules can easily be applied in designing supermarket shelves. Negative associations, on the other hand, such as $X \Rightarrow \neg Y$, where $\neg Y$ represents the negation (absence) of $Y$, might also be discovered through association mining. The negative associations are rarely used in application domains due to the lack of universally accepted methods for finding them.

The pioneering work of Agrawal et al. (AIS93) opened a flood of research in association mining. Since its publication, many algorithms are developed to conduct association mining. Association mining has contributed to many developments in related application areas such as product recommendation (Dem04). Recent developments have positioned association mining as one of the most popular tools in retail analytics as well (BSVW99). Traditionally, association mining generates positive association rules that reveal complementary effects. In other words, the rules suggest that purchasing some item(s) can generate the sales of (an)other item(s). Association mining can also be used to find so-called "Halo effects", where reducing the price of an item can increase the sales of (an)other item(s). Although positive associations are an integral part of the retail analytics (BSVW99), negative associations are rarely seen in application. However, they might be very useful to identify the substitution effects in a retail environment to understand "cannibalization" effects where reducing the price of an item adversely decreases the profitability of another item. By substitution we mean that a product is preferred and is purchased instead of another one. There are four main reasons that a substitution might occur between two products: Product Taxonomy, stock-out, price, product life cycle.

Clustering is a discovery process that groups a set of data points such that the intracluster similarity measure is maximized and the intercluster similarity measure is minimized (HKKM97). The discovered clusters are used to explain the characteristics of the data distribution in general. Unsupervised learning nature of clustering may result in certain unwanted properties in the resulting clusters. To avoid such situations, constraints can be introduced to the discovery process in order to form better clusters (BBD00; WC00; WCRS01). Constrained clustering is discussed and successfully applied on several different problems in the edited book (BDW08) by several authors including (LLJB08; DBB08; LL08; PRB08). Two types of constraints are common in (BDW08) namely must-link and cannot-link. These pairwise constraints determine whether the paired objects should be placed into the same or into the different clusters respectively.

The primary purpose of this paper is to develop a methodology to form product groups that the products within each group do not adversely *cannibalize* each other when the markdown optimization algorithm is applied. For this purpose, complementary products can be placed into the same cluster while the substitute products can be placed into the different clusters. In order to implement this idea, both positive and negative association mining can be used first to find complementary and substitute products respectively. Positive and negative item associations then can be considered as must-link and cannot-link constraint under the constrained clustering framework. Therefore a constrained clustering approach can easily be implemented to group the products of an apparel retailing firm. Hence there are two types of datasets involved. The first one is a transaction dataset from which the item associations can be derived. The second one is used for the clustering purposes in which the item (product) properties are contained e.g. weekly sales figures throughout the season.

The remaining of the paper is organized as following. In Section 2, we present an overview of the basic concepts in related studies through a concise review of relevant literature. In Section 3, we motivate and define our methodology with an illustrative example. We apply the methodology on a real dataset from an apparel firm and demonstrate its applicability in Section 4. Finally, we summarize our work and discuss future directions in Section 5. Throughout the paper the terms "item" and "product" are used interchangeably.

## 2 Related Work

Association rules are used in (HKKM97) to cluster related items as in the form of additional information. In (HKKM97), the frequent item-sets are used to construct a weighted hypergraph, and then a hypergraph partitioning algorithm is used to find the clusters. In (HKKM97), it is shown that clustering using association rule hypergraphs holds great promise in several application domains. The clustering method in (HKKM97) uses some similarity criteria related to the confidence of the association rules derived from frequent item-sets. Once items are clustered properly, then transactions can also be assigned to particular transaction clusters based on the similarity to item clusters by implementing a scoring scheme (HKKM97).

Conventional association mining mainly focus on positive associations and there exist many applications on this problem. As discussed above Apriori algorithm (AIS93) is the most common algorithm in this field. Therefore we skip the detailed information on how to find the positive association but we focus on negative ones. In (TKK01) authors propose a method of finding negative association via indirect associations. Figure 1 depicts such an indirect association $\{BC\}$ via item $A$. In Figure 1 itemsets $\{AB\}$ and $\{AC\}$ are both frequent. However, the itemset $\{BC\}$ is not listed as a frequent itemset in the association

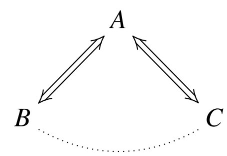


Figure 1. Indirect Association

mining results. The itemset $\{BC\}$ is said to have an *indirect association* via item $A$ and thus it is considered as a candidate negative association. Item $A$ in this case is called as a *mediator* for the itemset $\{BC\}$. 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.

## 3 The Methodology

The original k-means clustering problem can be posed as an optimization problem in the following way. Given a dataset $\mathcal{X} = \{x_i\}_{i=1}^n$ of $n$ points in $\mathbb{R}^d$ and a number $k$ of desired clusters, the problem is to find cluster centers $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ in $\mathbb{R}^d$ such that the sum of the 2-norm distance squared between each point $x_i$ and its *nearest* cluster center $\mathcal{C}_h$ is minimized (DBB08). Specifically:

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \sum_{i=1}^{n} \min_{h=1,\ldots,k} \left( \frac{1}{2} \|x_i - \mathcal{C}_h\| \right). \tag{1}$$

By (BMS97, Lemma 2.1), (1) is equivalent to the following problem where the min operation in the summation is removed by introducing “assignment” variables $Y_{i,h}$, $i = 1, \ldots, n$, $h = 1, \ldots, k$. Note that $Y_{i,h} = 1$ if data point $x_i$ is closest to center $\mathcal{C}_h$ and zero otherwise.

$$\begin{array}{ll} \underset{\mathcal{C},Y}{\text{minimize}} & \displaystyle\sum_{i=1}^{n} \sum_{h=1}^{k} Y_{i,h} \cdot \left( \frac{1}{2} \|x_i - \mathcal{C}_h\| \right) \\ \text{s.t.} & \displaystyle\sum_{h=1}^{k} Y_{i,h} = 1,\ i = 1, \ldots, n, \\ & Y_{i,h} \geq 0,\ i = 1, \ldots, n,\ h = 1, \ldots, k. \end{array} \tag{2}$$

There exist efficient network algorithms to solve the problem defined in Equation 2 (DBB08). Must-link and cannot-link constraints as well as the minimum cluster size constraints can be introduced to the clustering problem defined in Equation 2 as hard constraints without any difficulty as in the following problem formulation.

$$\begin{array}{ll} \underset{\mathcal{C},Y}{\text{minimize}} & \displaystyle\sum_{i=1}^{n} \sum_{h=1}^{k} Y_{i,h} \cdot \left( \frac{1}{2} \|x_i - \mathcal{C}_h\| \right) \\ & \displaystyle\sum_{h=1}^{k} Y_{i,h} = 1,\ i = 1, \ldots, n, \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} Y_{i,h} \geq \tau_h,\ h = 1, \ldots, k \\ & Y_{i,h} = Y_{j,h}\ i, j \in \{c_{=}(i,j)\}\ h = 1, \ldots, k \\ & Y_{i,h} + Y_{j,h} \leq 1\ i, j \in \{c_{\neq}(i,j)\}\ h = 1, \ldots, k \\ & Y_{i,h} \geq 0,\ i = 1, \ldots, n,\ h = 1, \ldots, k. \end{array} \tag{3}$$

where the set $\{c_=(i,j)\}$ $i,j=1,\ldots,n$ represents the must-link constraints and the set $\{c_{\neq}(i,j)\}$ $i,j=1,\ldots,n$ represents cannot-link constraints. However, if must-link and cannot-link constraints are introduced as hard constraints as in the problem in Equation 3, it is very likely that the underlying clustering problem will have no feasible solution at all for the large problems depending on parameter selection. This is not a desired outcome from a clustering algorithm. The algorithm should be able to produce cluster centers with as few constraint violations as possible. As in (LLJB08; LL08; PRB08), Lagrangian relaxation of the optimization problem defined in Equation 3 can be formulated as follows.

$$
\begin{array}{ll}
\underset{\mathcal{C},Y}{\text{minimize}} & \displaystyle\sum_{i=1}^{n}\sum_{h=1}^{k} Y_{i,h}\cdot\left(\frac{1}{2}\|x_i-\mathcal{C}_h\|\right)+\frac{\lambda^+}{2}\sum_{i,j\in\{c_=(i,j)\}|(Y_{i,h}\neq Y_{j,h}),h=1,\ldots,k}\mathbf{1} \\
& \displaystyle +\lambda^-\sum_{i,j\in\{c_{\neq}(i,j)\}|(Y_{i,h}+Y_{j,h}==2),h=1,\ldots,k}\mathbf{1} \\
\text{s.t.} & \displaystyle\sum_{h=1}^{k} Y_{i,h}=1,\ i=1,\ldots,n, \\
& \displaystyle\sum_{i=1}^{n} Y_{i,h}\geq\tau_h,\ h=1,\ldots,k \\
& Y_{i,h}\geq 0,\ i=1,\ldots,n,\ h=1,\ldots,k.
\end{array}
\tag{4}
$$

where $\lambda^+$ and $\lambda^-$ are penalty parameters for the respective constraint violations in Lagrangian relaxation. The constraint violations are represented as a logical value where $Y_{i,h}\neq Y_{j,h}$ case is a violation of a must link and $Y_{i,h}+Y_{j,h}==2$ case is a violation of cannot-link constraint considering the decision variable, $Y_{i,h}$, is a **0/1** variable. Notice that the parameter $\tau_h$, $h=1,\ldots,k$ is used for the minimum cluster size constraint. In other words, the cluster $h$ is required to have at least $\tau_h$ points in it. This constraint is introduced to ensure adequate number of points to fall in each cluster to form multi-product groups. Thus with this constraint, undesired empty clusters, single item clusters and clusters with very few points in conventional k-means clustering are avoided at the end of the clustering process.

The methodology proposed in this paper can be summarized in the following steps. Notice that the transaction dataset $\mathcal{D}$ is used for the association mining step and the constrained clustering is run on a different set of data, $\mathcal{X}$, in which the product properties can be represented.

1. **Determine Must-link Constraints.** Apriori algorithm is run on transaction dataset $\mathcal{D}$ to generate two-frequent itemsets with a minimum support value of *min_sup* and to determine the set of must-link constraints i.e. $\{c_=(i,j)\}$ $i,j=1,\ldots,n$.
2. **Determine Cannot-link Constraints.** Indirect association mining algorithm is run on $\mathcal{D}$ to generate itemsets that have pairwise *negative* association and to determine the set of cannot-link constraints i.e. $\{c_{\neq}(i,j)\}$ $i,j=1,\ldots,n$.
3. **Run the Constrained Clustering.** Do the following steps to generate product clusters until the solution converges

   A. **Cluster Assignment.** Let $Y_{i,h}^t$ be a solution to the Equation 4 with $\mathcal{C}_{h,t}$ fixed.

   B. **Cluster Update.** Update $\mathcal{C}_{h,t+1}$ as follows:

$$
\mathcal{C}_{h,t+1}=\begin{cases}\dfrac{\sum_{i=1}^{n}Y_{i,h}^t x_i}{\sum_{i=1}^{n}Y_{i,h}^t} & \text{if } \sum_{i=1}^{n}Y_{i,h}^t>0, \\ \mathcal{C}_{h,t} & \text{otherwise.}\end{cases}
\tag{5}
$$

   C. $t\leftarrow t+1$

Table 1. Pairwise Constraints for the Sample Data

| Must-link Constraints | Cannot-link Constraints |
|---|---|
| (Brown Bat, Silver Hair Bat) | (Wolf, Bear) |
| (Brown Bat, Pigmy Bat) | (Wolf, Badger) |
| (Brown Bat, House Bat) | (Bear, Badger) |
| (Fur Seal, Grey Seal) | |

### 3.1 An Illustrative Example

A sample dataset that contains the numbers of different kinds of teeth for a variety of mammals from SAS Support web site[1] (available also at an alternative url[2]) is used in this section to illustrate our approach at a small scale. This particular dataset is chosen to depict clear examples of pairwise relationships. Thirty two mammals (e.g. Brown Bat, Rabbit, Raccoon and Elephant Seal) have their eight different teeth type characteristics in this particular dataset. Some of these mammals come from the same familia thus must-link constraints can be constructed accordingly. Obviously, some of them belong to the different familia where cannot-link constraints can easily be constructed as well. The full hierarchical clustering of the sample data can also be found at the data web site which is found by SAS clustering procedures.

Must-link and cannot-link constraints introduced for this sample problem is listed in Table 1. These constraint are selected arbitrarily for the illustration purposes. Notice that the four different types of bat must be placed into the same cluster. This shows that the pairwise constraints are expressive enough to represent more complex relations. For the illustration purpose, a chain of three pairwise negative constraints are introduced in our implementation as specified in Table 1. These constraints simply indicate that the three mammals, namely wolf, bear, and badger, should be placed into the different clusters. Therefore at least three clusters are needed (*3*-means) for a feasible solution with the hard constraints as defined in Equation 3. If the Lagrangian relaxation is used instead, the algorithm will be able to find a solution by letting the violation of at least one constraint. In Figure 2, the comparison of the regular k-means and the constrained k-means clustering with Lagrangian relaxation are compared in terms of their objective values while the number of clusters, *k*, is varied. The minimum cluster size is set to four for all of these clusters. Thus the number of clusters vary between two and eight in Figure 2. The parameters $\lambda^+$ and $\lambda^-$ for the Lagrangian relaxation are set to 100.

For the two-cluster case Lagrangian relaxation is able to cluster objects with single constraint violation. It can be observed from Figure 2 that the regular clustering algorithm gets stuck at local solutions when the number of clusters are five and eight. The algorithm finds solutions that have higher objective values than the constrained clustering. Local solutions are common for the *k*-means algorithm. That's why, introducing appropriate constraints regarding the cluster formation can definitely improve the resulting cluster quality.

The implementation of our methodology is done by using IBM ILOG OPL Studio. The model and data files of the sample problem are available at the first author's web site[3]. For those who are interested in this application, IBM ILOG OPL Studio can be downloaded for free of charge for the academicians at IBM Academic Initiative web site[4].

## 4 Experimenting with Retail Dataset

In this section the applicability of the methodology is demonstrated on real-life data, specifically the retail sales data originating from an apparel chain. Store level data coming from the largest apparel retailer in Turkey have been used in this work. In our study, the complete sales, stock and transshipment data that belong to a single merchandise group (men's clothes line) collected from all of the stores (over 200 of them across the country) during the summer season of year 2007 were available.

[1] http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/statug_tree_sect018.htm
[2] http://tinyurl.com/y7pyrw2
[3] www.sakarya.edu.tr/∼ademiriz/SakaryaWebSite/Const_Clus.zip
[4] https://www.ibm.com/developerworks/university/academicinitiative/index.html

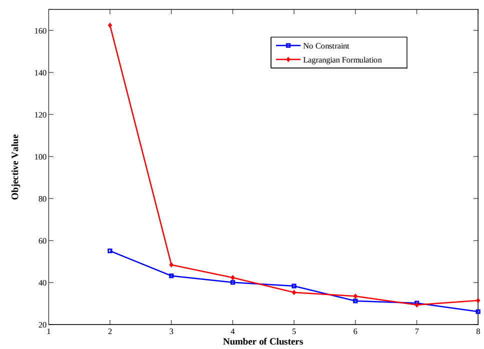


Figure 2. Objective Values of Regular and Constrained k-means Clustering by Varying $k$

$$\texttt{Merch. Group} \longrightarrow \texttt{Sub} - \texttt{Merch. Group} \longrightarrow \texttt{Category} \longrightarrow \texttt{Model} \longrightarrow \texttt{SKU}$$

Figure 3. A Typical Representation of the Product Hierarchy

A typical product hierarchy for an apparel retailer can be depicted as in Figure 3. The products at the Stock Keeping Unit (`SKU`) level are the ones purchased by customers directly in this hierarchy. At that level each color and size combination that belongs a `model` is assigned to a unique SKU. A `category` is composed of similar models e.g. long sleeve shirts. A `merchandise group` can represent the whole product line for a gender and age group e.g. women's clothes line. A `sub-merchandise group` further divides this large group. Notice that this particular product hierarchy is just a typical representation and the product hierarchy may change from one firm to the another one.

Since the SKU level store data show high variability, the data aggregated at the model level are used, which is the immediate parent of the SKU level. Positively and negatively associated item pairs are determined by using transaction data, $\mathcal{D}$. There are 2,753,260 transactions and 4,376,886 unique items sold in the transaction data with 716 products (model). Therefore the sparsity of the data (Dem04) is approximately 99.74% which is very high. Notice that the transaction data include the single-item transactions.

After conducting positive association mining with a minimum support count of 100 by using SAS Enterprise Miner, 3930 two-itemsets are found to be frequent. We only use the frequent two-itemsets in our study to represent the frequent pairs. Thus we do not directly include the association rules in our analysis. The top-5 frequent pairs have support counts of 22131, 17247, 17155, 14224, and 11968 respectively, within the 2,753,260 transactions. The frequent pairs are used for finding the negatively related pairs via indirect association mining which has yielded 5,386 negative itemsets including a mediator item. The resulting negatively associated pairs forms a set of 2,433 unique pairs without a mediator. The negative associations can be found by indirect associations (TKK01) which is essentially composed of three-step SQL join operations implemented also in SAS.

After generating pairwise positive and negative associations, it can be observed that the number of the constraints is considerably high considering the underlying optimization is a mixed integer programming (MIP) model and logical variables are used to represent Lagrangian relaxation by the solver. On the other hand, the original support count, 100, may be considered as relatively small number. It might be reasonable to consider only

Table 2. Objective Values and Run Times of Clustering Methods

| Number of Clusters | Constrained Clustering | | Regular Clustering | |
|---|---|---|---|---|
| | Objective Value | Run Time (min.) | Objective Value | Run Time (min.) |
| 10 | 427,071,984 | 91 | 855,445 | 4 |
| 20 | 653,048,052 | 196 | 751,448 | 9 |
| 30 | 874,050,706 | 110 | 702,268 | 19 |

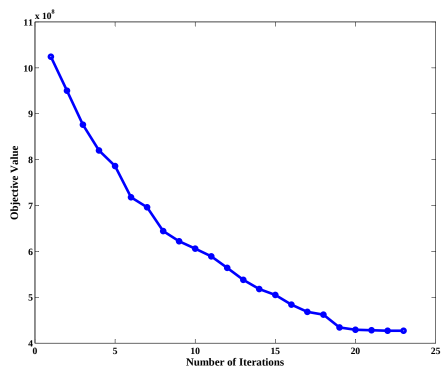


Figure 4. Convergence of Lagrangian Relaxation of Constrained Clustering ($k$=10)

pairwise item associations that those have a support count of at least 400. In this case, there are 600 positive associations (must-link constraints). There are 227 distinct items within these 600 pairs. In a way, these items are prioritized based on their support counts. These distinct items have 1073 negative pairwise associations (cannot-link constrainst) among themselves. Therefore the size of the constraints can be reduced significantly with this approach to 600 and 1073 for the must-link and cannot-link constraints respectively.

For the scope of this study, weekly sales figures during the summer season (spans more than half of the year) are used as the characteristics of the products. Additional properties (including the categorical ones) can be incorporated into the clustering process. However data from thirty-four weeks period are used for the 716 products in the clustering phase. Notice that the data matrix, $\mathcal{X}$, has a dimension of $716 \times 34$. This is a relatively small dataset considering the efficiency of the conventional k-means clustering algorithms. The k-means algorithm is run on this data with constraints and without constraints separately. The number of clusters, $k$, is set to 10, 20, and 30 during the experiments. The results reported in Table 2 have the actual run times in minutes in addition to objective values found by the optimization software. The experiments are run on a dual-processor quad-core Xeon machine with 6GB memory on 64 Bit Windows Server 2008 operating system. The parameters $\lambda^+$ and $\lambda^-$ for the Lagrangian relaxation are set to 1,000,000. A representative convergence result is depicted in Figure 4.

The results in Table 2 indicate that many number of constraints are violated and the results are more likely local solutions (minima). Considering the value of the penalty terms $\lambda^+$ and $\lambda^-$, we can conclude that approximately 425 must-link and cannot-link constraints are violated in the experiment when $k$ is set to 10. Similarly approximately 650 and 870 of them are violated when the number of clusters, $k$, is set to 20 and 30 respectively. We observe from our unreported experiments that the cannot-link constraints are harder to satisfy. Since the regular k-means clustering problem can be solved by the efficient network problem solver in CPLEX, it takes considerably less time to run it. However, the constrained clustering problem is a MIP model and it takes significantly longer to solve it.

## 5 Discussion and Conclusion

The work developed in this paper is a part of a larger revenue management research project in which product sales are forecasted within each cluster in a multivariate sense. Some experimental results from the constrained clustering application are reported in this paper. Nevertheless, a novel methodology is proposed to generate multi-product groups to run markdown optimization. The methodology is designed based on a MIP model.

An alternative method to form multi-product groups would be the usage of regular k-means algorithm and then filtering the clustering results with the outcome of the positive and negative association mining. By filtering we mean that only those products that satisfy the must-link and cannot-link constraints remain in their prospective clusters otherwise they are removed. However, this quick solution does not guarantee the assignment of each product to a cluster. Many products will be left unassigned depending on the positive and negative associations. The constrained clustering methodology proposed in this paper satisfies as many constraints as possible and the resulting clusters are expected to be more plausible.

## REFERENCES


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.

P. S. Bradley, K. P. Bennett, and A. Demiriz. Constrained k-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, May 2000.

S. Basu, I. Davidson, and K. L. Wagstaff, editors. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman and Hall/CRC, 2008.

Z. Bulut, Ü. Gürler, and A. Şen. Bundle pricing of inventories with stochastic demand. *European Journal of Operational Research*, 197:897–911, 2009.

P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 368–374, Cambridge, MA, 1997. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-03.ps.Z.

Tom Brijs, Gilbert Swinnen, Koen Vanhoof, and Geert Wets. Using association rules for product assortment decisions: a case study. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 254–260, New York, NY, USA, 1999. ACM.

A. Demiriz, K. P. Bennett, and P. S. Bradley. Using assignment constraints to avoid empty clusters in k-means clustering. In Basu et al. (BDW08), pages 203–222.

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

Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Clustering based on association rule hypergraphs. In *Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97)*, 1997.

Z. Lu and T. K. Leen. Pairwise constraints as priors in probabilistic clustering. In Basu et al. (BDW08), pages 59–90.

T. Lange, M. H. Law, A. K. Jain, and J. M. Buhmann. Clustering with constraints: A mean-field approximation perspective. In Basu et al. (BDW08), pages 91–122.

M. K. Mantrala and S. Rao. A decision-support system that helps retailers decide order quantities and markdowns for fashion goods. *Interfaces*, 3(Part 2 of 2):S146–S165, May-June 2001.

R. G. Pensa, C. Robardet, and J. F. Boulicaut. Constraint-driven co-clustering of 0/1 data. In Basu et al. (BDW08), pages 123–148.

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

K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1103–1110, 2000.

K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 577–584, 2001.