

[This document contains the author's accepted manuscript. For the publisher's version, see the link in the header of this document.]

A New Heuristic for Learning Bayesian Networks from Limited Datasets: A Real-Time Recommendation System Application with RFID Systems in Grocery Stores

By Esma Nur Cinicioglu and Prakash P. Shenoy

Istanbul University School of Business and University of Kansas School of Business

Paper citation:

Shenoy, Prakash. A New Heuristic for Learning Bayesian Networks from Limited Datasets: A Real-Time Recommendation System Application with RFID Systems in Grocery Stores. *Annals of Operations Research*.

Keywords:

Bayesian networks, Heuristic for Bayesian networks, RFID, Real-time recommendation systems, Targeted advertising

Abstract:

Bayesian networks (BNs) are a useful tool for applications where dynamic decision-making is involved. However, it is not easy to learn the structure and conditional probability tables of BNs from small datasets. There are many algorithms and heuristics for learning BNs from sparse datasets, but most of these are not concerned with the quality of the learned network in the context of a specific application. In this research, we develop a new heuristic on how to build BNs from sparse datasets in the context of its performance in a real-time recommendation system. This new heuristic is demonstrated using a market basket dataset and a real-time recommendation model where all items in the grocery store are RFID tagged and the carts are equipped with an RFID scanner. With this recommendation model, retailers are able to do real-time recommendations to customers based on the products placed in cart during a shopping event.

A New Heuristic for Learning Bayesian Networks from Limited Datasets: A Real-Time Recommendation System Application with RFID Systems in Grocery Stores

Esma Nur Cinicioglu^a and Prakash P. Shenoy^b

^a Istanbul University School of Business, 34320 Avcilar, Istanbul, Turkey

^b University of Kansas School of Business, 1300 Sunnyside Ave, Summerfield Hall, Lawrence, KS 66045-7601, USA

<esmanurc@istanbul.edu.tr>, <pshenoy@ku.edu>

Abstract

Bayesian networks (BNs) are a useful tool for applications where dynamic decision-making is involved. However, it is not easy to learn the structure and conditional probability tables of BNs from small datasets. There are many algorithms and heuristics for learning BNs from sparse datasets, but most of these are not concerned with the quality of the learned network in the context of a specific application. In this research, we develop a new heuristic on how to build BNs from sparse datasets in the context of its performance in a real-time recommendation system. This new heuristic is demonstrated using a market basket dataset and a real-time recommendation model where all items in the grocery store are RFID tagged and the carts are equipped with an RFID scanner. With this recommendation model, retailers are able to do real-time recommendations to customers based on the products placed in cart during a shopping event.

Keywords: *Bayesian networks, Heuristic for Bayesian networks, RFID, Real-time recommendation systems, Targeted advertising*

1 Introduction

In the last 10 years, after welcoming e-commerce to our everyday lives, the traditional way of shopping has changed tremendously. With e-commerce, we have the comfort of shopping from home, can select from a large variety of products, and can do some comparison shopping. These features of e-commerce have altered the long-held shopping habits of customers. On the other

side, retailers are enjoying the benefits that come along with e-commerce—going beyond brick and mortar stores, having access to customers all over the world, achieving considerable savings in operational costs, etc. But above all, e-commerce has enabled retailers to obtain information on shopping habits of customers that they have wanted for many years. In e-commerce, retailers are capable of identifying which products a customer is potentially interested in based on the products the customer has placed inside his/her virtual cart or based on the products that the customer has just clicked on. And by analyzing that information, they are able to do real-time recommendations to customers. Unlike the methods of traditional shopping to affect customers' purchase decisions (giving coupons at the checkout, analyzing past purchases, installing kiosks etc.), in e-commerce the retailers are able to influence the decision-making process of the customer in real-time as the customer is shopping.

Addressing the issue described above, Cinicioglu *et al.* (2007) emphasized the need to have real-time recommendation systems in traditional grocery stores. As a solution, they suggest the use of RFID systems in grocery stores, where each product is tagged with a RFID, and there is a RFID scanner attached to each cart allowing the system to identify the products that a customer has placed inside the cart during a shopping event. With this new model, retailers are able to do targeted advertising to customers based on the information of the products placed in cart during shopping events. This model uses Bayesian networks, which with its ability to learn predictive relationships between the variables, forms a perfect tool as a real time recommendation system. To illustrate this model, a random selection from a dataset available for the Netflix prize competition¹ was used where they converted the Netflix dataset with movies as the products instead of grocery items. The Netflix dataset is based on customers' movie ratings, which is collected through the voluntary participation of the customers. For that reason, during the conversion of the Netflix dataset into a grocery dataset, the authors had to make an assumption that the movies not rated by the customer were movies that were not in customer's cart. When it came to apply the same model with a real grocery dataset (in order to eliminate the assumption made in the previous research), the authors were faced with the fact that learning Bayesian networks from limited datasets and making useful recommendations using the learned network is

¹ The Netflix prize competition seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on the ratings of the movies they have already seen.

quite problematic. We call the grocery dataset limited since, despite the huge number of product variety contained, this dataset is sparse in the sense that a tiny portion of the products tend to constitute a big part of the whole dataset. In this research, we develop a new heuristic for learning Bayesian networks from a sparse dataset to reveal the information hidden inside the dataset to form some useful recommendations. This heuristic is demonstrated using a retail market basket dataset, supplied by an anonymous Belgian retail supermarket store (Brijs *et al.*, 1999).

The outline of the remainder of the paper is as follows. In Subsection 2.1, we provide a brief introduction to recommendation systems. This is followed with an introduction of Bayesian networks as a dynamic recommendation system in Subsection 2.2. A review of the current Bayesian network learning algorithms and heuristics is also provided in the same section. In Section 3.1, a futuristic scenario of a grocery store model is introduced, where Bayesian networks are used as a recommendation system. In Section 3.2, the details of the grocery dataset used in this research are provided. This grocery dataset forms a perfect example of a limited data structure. In Section 4.1, we introduce a new heuristic for learning Bayesian networks from limited datasets. This new heuristic is described and its performance is demonstrated via the use of the grocery dataset described in the previous section. In Section 4.2 we evaluate the proposed heuristic by applying it to modified versions of the original data set and validate its success. In Section 5, a case study is depicted to illustrate the model developed. Finally, in Section 6, we summarize and conclude.

2.1 Recommendation Systems

The idea of mass customization can be summarized as developing multiple products that meet multiple needs of multiple customers at a minimum cost (Pine, 1993). As a consequence of this idea, there is a huge variety of products offered in grocery stores. The information overload created by this enormous product variety, and the inability of customers to identify the differences between the products, gave the retailers the incentive to develop new methods to inform the customers about the products to be able to affect customer's purchase decisions. All of these methods—such as newspaper ads, reconfiguration of shelves, having information kiosks, giving discount coupons at checkout—as a recommendation system, they lack the ability to influence the decision-making process of the customer in real time as the customer is shopping.

Looking to e-commerce on the other side, we see that with the ability to collect real time information about customer's shopping behavior, e-commerce sites are capable of making real-time recommendations to customers and hence can influence the customer's purchase decisions. Additionally, in e-commerce, by applying mass customization principles to the presentation of products in the online store, the recommendation systems can address the big information overload created by the huge product variety (Pine and Gilmore, 1999).

The recommendation systems, depending on the technology used, can be classified into two classes: Content-based filtering, and collaborative filtering. Collaborative filtering, first introduced by Resnick *et al.* (1994), is defined as predicting the preferences of an active user given a database of preferences of other users (Mild, 2003). The main difference between collaborative filtering and content-based filtering is that collaborative filtering does not rely on content descriptions of the items, but depends purely on preferences expressed by a set of users (Yu *et al.*, 2004). In e-commerce, collaborative filtering is widely used as a tool for targeted advertising. Its basic assumption is that a good way to find interesting content is to find other people who have similar interests, and then recommend items that those similar users like (Resnick and Varian, 1997). Collaborative filtering can be classified into two general classes of algorithms, memory-based and model-based. In memory-based algorithms, the entire user database is used to make predictions. In contrast, model-based algorithms use the user database to estimate (or learn) a model, which is then used for predictions (Breese, 1998). Bayesian networks (BNs) are an example of models learned from user databases that can be used for predictions. We will describe BNs in the following section.

2.2 Bayesian Networks

A Bayesian network is a directed acyclic graph where nodes represent random variables, and the directed arcs represent dependencies between the variables. If there is a directed arc from a variable X_1 to a variable X_2 , then we call X_1 a *parent* of X_2 , and X_2 a *child* of X_1 . Each variable in a Bayesian network is associated with a conditional probability distribution given its parents, and the product of these conditional probability distributions (also called conditional probability tables or CPTs in short) constitute the joint probability distribution of the variables in the network:

$$P(X_1, \dots, X_N) = \prod_{i=1}^N P(X_i | Pa(X_i)), \quad (1)$$

where $Pa(X_i)$ denotes the set of parents of X_i .

BNs are well suited for a recommendation system. Using BNs, we can learn the relationships between the variables. These relationships constitute the joint probability distribution for all variables as stated in Equation (1). Making inferences in BNs means finding posterior marginals of variables of interest based on observations of other variables. Using a BN, we can identify the products that a customer will be interested in (and which are likely to be purchased by the customer) based on the information of the products placed by the customer in a cart. Thus, this makes BNs a good tool for a recommendation system. For data mining, BNs have the capacity to handle incomplete data, ability to learn causal relationships, and can combine information from domain knowledge and datasets (Heckerman, 1997). Friedman *et al.* (1997) attributes the growing interest for learning BNs from data to semantic clarity and understandability of BNs by humans, the ease of acquisition and incorporation of prior knowledge, and the ease of integration with optimal decision-making models. In general, there exist two BN learning algorithms: Constraint-based, and search & score methods. Constraint-based methods use statistical or information theoretical measures to estimate (from the data) whether certain conditional independencies hold between variables (Liu *et al.*, 2007). The path condition (PC) algorithm is an example of constraint-based method. Constraint-based methods are criticized for their sensitivity to failures in conditional independence tests. In a search & score method, a statistically motivated score is defined which measures the fitness between a graph and the dataset. A structure that has the maximum score is selected as the learned network. A problem with the search & score method is that the cost of the evaluations are acute when it comes to massive datasets (Friedman *et al.*, 1999).

Bayesian networks typically have smaller memory data requirements and allow for faster predictions than a memory-based technique such as correlation (Breese, 1998). Oniško *et al.* (2001) report, that when sufficient amount of data is available, they can be used to learn both the structure and the parameters of a Bayesian network model. However, in case of small datasets, learned models can be of low quality. One reason for low quality is that in small datasets many conditioning cases are represented by too few or no data records. One approach to handle insufficient data is to use credal networks. Unlike BNs, variables in credal networks are

associated with probabilistic inequalities. But a key problem associated with credal networks is making inferences, i.e., generating bounds on posterior probabilities and posterior expectations (Cozman, 2000).

Another example where Bayesian networks have limitations is the case of limited datasets with high dimensions (Liu *et al.*, 2007). In terms of data mining literature this is a problem of imbalanced data sets. Many methods to deal with imbalanced data sets are proposed in data mining. Some of them involve techniques intended to achieve a more balanced data set, such as random-over sampling or random under-sampling. Both of these methods are associated with known-drawbacks such as the elimination of useful data and increased likelihood of over-fitting. A detailed review can be found in (Gu *et al.*, 2008).

Considering the late popularity of learning Bayesian networks from data, the problem of limited data sets with high dimensions is still new for BN community. Some of the approaches suggested in literature are as follows: Goldenberg and Moore (2004) question whether it is useful to learn a Bayesian network structure with thousands of nodes, and proposes to use frequent sets to speed up the structural search in sparse datasets (such as supermarket transactions). But learning Bayesian networks from data, particularly in the context of a recommendation system, also requires that the inference be useful for the customer. Along this line, Jaroszewics and Simovici (2004) suggest including users background knowledge in the process and to prune frequent item sets, accordingly. For instance, the necessary path condition (NPC) algorithm, which is an enhancement of the PC algorithm, allows the user to interactively decide on the directionality of the undirected links. The disadvantage of both PC and NPC algorithms is that their outcomes are structures and not conditional probability matrixes (Madsen *et al.*, 2004). Friedman *et al* (1999) suggest an algorithm that limits the search space for learning a BN by restricting the parents of each variable to belong to a small subset of candidates. Though this approach may achieve a faster learning process, it has some disadvantages. In data mining, it is often the case that each individual rule may not be interesting but a group of them together can represent an important piece of knowledge (Liu *et al.*, 2006). Thus, by restricting the parents of each variable to a small subset, we may miss higher-order interactions.

Although encouraging results are achieved in learning BNs from limited datasets, there is still the need for new heuristics and algorithms. Most of the existing algorithms to learn BNs from limited datasets are only concerned with the general quality of the learned BN. However,

for a dynamic recommendation system (such as the futuristic grocery store example that we are going to talk about in the following section), we are interested in the quality of the inferences made for recommendations. For that reason, there is a need for a new heuristic, which assists in the selection process of the variables included in the BN, so that better performing BNs may be obtained from sparse datasets. While the problem of variable selection has a huge literature in statistics, this is not the case for Bayesian networks. Besides, as discussed in Cui *et al.*(2010), traditional methods of stepwise variable selection do not consider the interrelations among variables and may not identify the best subset for model building.

In subsection 3.1, we first give the details of the real-time recommendation system of the grocery store model using RFID systems. In subsection 3.2, we describe the grocery dataset that is used for the analysis.

3.1 Model: Using Radio Frequency Identification Systems (RFID) in a Grocery Store

Radio frequency identification (RFID) is a generic term for a variety of technologies that use radio waves to automatically identify individual items (Cavoukian, 2004). An RFID system consists of two basic parts: A tag and a reader. Readers, depending upon design and technology used, may be a read-only or a read-write device (Finkenzeller, 1999). They capture the information stored or gathered by the tag. The main difference of RFID systems over its preceding alternatives like barcodes is its ability to identify individual products from a distance. This distance may range from a few centimeters up to hundreds of meters depending on the type of technology used.

Though originally developed for the identification of friend/foe during World War II (Hicks, 1999), the real popularity of RFID systems came after its potential for commercial applications is realized, thanks to its decreasing cost over the last ten years. The use of RFID systems in commercial applications has opened a new era for retailers in the sense that it allowed them to identify and track individual products through operations management activities. With this capability of RFID systems an improved inventory visibility can be obtained on the retailer's side, which may prevent theft, shrinkage, spoilage and also stockout problems. For these reasons, the worlds' largest retailer, Wal-Mart, began to require its top 100 suppliers to place RFID tags

at the case or pallet-level in 2005, which later increased to the top 600 suppliers in 2007 (Whitaker *et al.*, 2007). Moreover, since April 2003, Metro AG, the world's third-biggest retailer, is testing the use of RFID technology and other technology based systems in its Future Store Initiative (www.futurestore.org).

But above all, the real opportunity of RFID systems is that they provide a great source to collect massive amounts of data which researchers were never able to get hands on. In order to transform this massive amount of data into managerially useful information, we need original application models and innovative ways to process the newly obtained datasets.

In this research we use the futuristic scenario described in Cinicioglu *et al.* (2007), where all items in the grocery store are tagged with RFIDs, and the grocery carts are equipped with a RFID scanner. On each cart there is also a small screen where promotions are displayed. To demonstrate the new heuristic that is developed in this research, a retail market basket dataset, supplied by an anonymous Belgian retail supermarket store (Brijs *et al.*, 1999), is used. Using the ability of RFID systems to identify individual items through a distance we can get the information of the products that the customer is placing in the cart, and using that information, we can identify those products on promotion that are more likely to capture the customer's interest. Using BNs, we can learn the predictive relationships between the variables; hence we can identify the products (based on the information of the products placed in cart) in which the customer will be interested and which are likely to be purchased by the customer. One useful aspect of Bayesian networks is that the observed evidence can easily be integrated to the inference process. Hence, with each product that a customer is placing in the shopping basket, we can update our network resulting with new suggestions for the customer. An important advantage of this model is that promotions and advertisements happen in real time during a shopping event, so that it may have a direct influence on customer's decision-making process. Use of real grocery data allows us to eliminate the assumption made in Cinicioglu *et al.* (2007) that the movies not rated by the customer were movies that were not in the customer's cart. WinMine (Heckerman *et al.*, 2000), a tool developed by Microsoft Research, is used to learn Bayesian networks. The following section gives details of the retail market basket dataset used in this research.

3.2 Retail Market Basket Dataset

The dataset used in this research is a retail market basket dataset, supplied by an anonymous Belgian retail supermarket store (Brijs *et al*, 1999). The data has been collected over three non-consecutive periods, resulting in approximately 5 months of data. Although this may sound reasonable to contain enough data, a detailed investigation on the structure of data reveals its limited composition.

Each line in the dataset corresponds to a shopping basket. The distinct items in the dataset are encoded by numbers², but the information to which products those numbers correspond is not provided in the dataset. We also do not know how many of the same product is present in a single shopping basket. The dataset contains the product information of 51,273 different shopping baskets but this only refers to 5,133 different customers. The number of distinct products included in the dataset is 14,472 but the majority of products are only present in a few baskets whereas five products with the highest purchase frequencies amount to about 16% of the whole dataset, which shows a sparse data structure with high dimensions.

Depending on the purpose of a single grocery shopping event, the products selected may tend to belong to the same grouping. In this context, aggregation techniques might help for a better analysis, however, the limited structure of the grocery dataset described above makes the use of cluster analysis unsuitable in this case.

Remember for our analysis, we are only interested to know whether a product is present in the shopping basket or not. The quantity of the products purchased does not matter for us. In order to achieve efficiency in our analysis, we transformed the dataset into a form where every single row in the dataset gives information about one shopping basket and one product that is placed inside. Thus, the same basket can be seen in as many rows as the number of products in it. In the following section, we describe the new proposed heuristic, and evaluate it via the use of the grocery dataset described in this section.

² Although most of the products are identified by a unique barcode, some article numbers in the dataset represent a group of products rather than an individual product item [Brijs *et al*, 1999].

4.1 A New Heuristic for Learning Bayesian Networks from Limited Datasets

In literature there exist various Bayesian network learning algorithms and many others for learning Bayesian networks from sparse datasets (as discussed in subsection 2.2). However, we needed to develop a new heuristic and the reason for it is as follows: This research with the proposed application given in the previous section has two requirements. First, we want to be able to learn BNs from sparse datasets (like the one described before). Second, we want the inferences made from the learned network to be useful for a recommendation system. Hence, the BN that we want to learn should be proficient to fulfill both of these requirements.

The procedure we followed is as follows: Learning Bayesian networks from datasets will be performed using WinMine program (Heckerman *et al.*, 2000), a software developed by Microsoft Research. There exist various programs for learning Bayesian networks. Using WinMine the conditional probability tables of the variables in the learned network can be obtained, which are used in the proposed heuristic. But the main incentive for the selection of WinMine program is its ability to automatically calculate log-scores and lifts over marginals of the learned BNs. Log-score is a quantitative criterion to compare the quality and performance of the learned BNs. Hence, the evaluation of the proposed heuristic will be made using the log-scores and lifts over marginals of the learned networks. For the calculation of the log-score, it is necessary to divide the dataset into a test and a training set. Using the WinMine Toolkit, a 70/30 train and test split is performed for all the BNs learned in this research and the accuracy of the learned model on the test set is then evaluated using the log score. The formula for the log score is given below

$$\text{Logscore}(x_1, \dots, x_N) = \sum_{i=1}^N \log_2 p(x_i | \text{model}) / nN, \quad (2)$$

where n is the number of variables, and N is the number of cases in the test set. Using WinMine we can also compare the difference between the log scores of the provided model and the marginal model. This difference is called lift over marginal log-score. A positive difference is desired between the provided model and the marginal model, signifying that the model outperforms the marginal model on the test set. In the same way that a regression model is more accurate than a simple baseline model chosen in the form of a mean dependent value, the “lift

over marginal” log-score provides information on how well the model fits the data (Scuderi & Clifton, 2005)

Using the entire dataset to construct the BN is not possible since the huge number of different products correspond to 14,472 variables in the BN. To learn a BN of that size, a much higher quantity of market basket data is necessary than it is available in this dataset. Hence, a fresh approach is needed to reveal the information contained in the dataset.

As discussed in section 2.2, one approach to handle sparse datasets with high dimensions is to use the frequent sets for the analysis, ensuring that enough data is available for all the variables included in the dataset. Following that approach, we can select the most frequent 30 products and learn the BN from that dataset. The corresponding log-scores are given in Table 1 below.

Table 1: The corresponding log-scores of the learned BNs

	log-score	Lift over marginal	Improvement obtained
most frequent 30	-0.186378	0.0195231	1.1813 %
most frequent 6 th -30 th	-0.207396	0.0114608	0.6907 %
most frequent 10	-0.322541	0.0642158	3.613 %
heuristic 10	-0.124822	0.229405	13.4829 %

The figures of log-scores and lift over marginals should be read as follows: The BN created with the data of the most frequent 30 products has a log-score of – 0.186378 which translates into a probability of 87.88% of correct prediction. A further investigation to determine the performance of the learned BN is made by calculating the lift over marginal, which is the difference between the log-scores of the provided and the marginal BNs. The marginal model uses the marginal probabilities of the products in the dataset, ignoring the information of the products that are placed in cart. If we ignored the products in cart and used the marginals for prediction, the average probability of correct prediction would be 86.6996%. Hence, with that BN model the improvement achieved is only 1.1813%.

As illustrated above, choosing the most frequent products for the analysis does not guarantee the performance and quality of the learned network and the reason for that is as follows: In grocery datasets it is almost always the case that some products like milk, eggs, and bread, are purchased on a regular basis by the customers. As a result, they occupy the top of the list as the most purchased products in the dataset, leaving a wide gap behind to the followers. A BN created from such a dataset, will naturally have no lift over marginal and this is an expected result, since

these products will have very high marginals. However, the main purpose of having a product recommendation system is to trigger the demand for some of the products inside the store. In that sense, making recommendations of milk or eggs for every possible kind of product placed in the shopping cart and overlooking some essential information, deteriorates the performance of the learned BN as also shown by the corresponding log-scores given in Table 1. It should be noted that, being purchased at the same shopping trip does not always necessarily show the tendency for those products to be purchased together by the customers. It may just be the result of high and uneven (compared to other products) purchase frequencies of those products that are purchased on everyday basis.

There is a huge gap between the frequencies of the top five most purchased products and the rest of the dataset. About 55% of the 14,472 distinct products in the dataset are present in only 10 baskets or less. In contrast, this number is 82,040 for the top five products in total³. The frequencies of the top 7 most purchased products are given in Table 2 below. In this table the figures provided in the third row indicate the proportional frequency of each of these products to the following most purchased one. As illustrated, this proportion rises up to 3.47⁴⁵ for the fifth most purchased product p_{32} , whereas it drops again, down to 1.196 for the product p_{65} . Anticipating that top five most purchased products are the ones bought on a daily basis (like milk, eggs etc.) we exclude them from the selection process of the products to create the BN.

Table 2: The frequencies and proportional frequencies of the top seven most purchased products

Product	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{89}
Frequency	29288	24201	10554	9100	8897	2562	2142
Proportion	1.21	2.29	1.16	1.023	3.47	1.196	1.088

But still, as selection of the most purchased products does not help for the analysis, so is the exclusion of the top 5. As given in Table 1, when the top 5 products are excluded and the remaining 25 most purchased are used to learn the BN, the improvement obtained even drops to 0.6907 %. The corresponding BN is shown in Figure 1 below.

³ This number is the sum of each of these products' count for the baskets they are in.

⁴ Rounded to two decimal places $8897 / 2562 = 3.47$

⁵ In this dataset 3.47 indicates the point where a big drop in the sales figures of the most purchased products happen

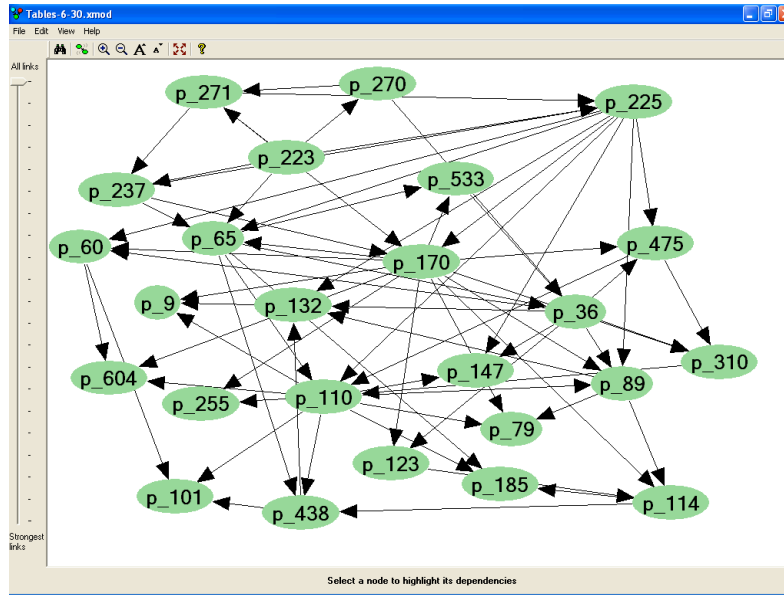


Figure 1. The BN created using the 6th-30th most purchased products

The purpose of this research is to learn Bayesian networks from sparse datasets and to achieve better performing Bayesian networks for recommendation systems than the ones created from raw data without the application of the proposed heuristic. With the proposed application, we aim to make recommendations based on the products placed in cart. Hence, in order to achieve an accurate demonstration of a single grocery shopping the products used in the BN should be representative for a market basket, meaning that they are likely to be purchased on the same shopping trip. A Bayesian network representation includes conditional probability tables for each variable. We will use these conditional probability tables to see the associations between the products and to select the products to be used in the final BN⁶. The idea is to use this information contained in the BNs as a selection basis to identify the products that have high associations with each other; the ones that are likely to be purchased on the same shopping trip.

We proceed as follows: The first step of the heuristic is to select a certain number of products that are going to be used to create the BN. In this case, excluding the top five the dataset for the most purchased 25 products is used so we have enough data available to build the BN. This way we avoid resulting in poor recommendations as a consequence of the high and uneven frequencies of the daily-consumed products. Each variable possess two different states “0” and

⁶ Throughout the paper the “final BN” phrase will refer to the BN which is created on the last step of the application of the heuristic after the variables to be included in the BN are determined. This final BN constitutes the basis for the performance evaluation of the proposed heuristic.

“1”, “0” indicating that the product is not in cart and “1” indicating that the customer has placed the product inside the cart.

The main purpose of learning this BN is to obtain the information about the dependencies between the products and to learn the conditional probability tables. Using conditional probability tables we can observe the change on the purchase probabilities of a child node depending whether the parent products are present in cart or not. This change is high for some products, indicating that they are highly associated with their parents and low for some, as a result of weak associations.

Consider the CPT of the product p_{271} given in Table 3. The purchase probability of the p_{271} increases from 0.037 to 0.365 when p_{270} is included in the basket. The purchase probability of p_{271} increases even to 0.418 when the product p_{223} is added. These high changes in the purchase probabilities of p_{271} suggest that this is an example of a product that is highly associated with its parents.

Table 3: The CPT for p_{271}

CPT: p_{271}			
p_{223}	p_{270}	$(p_{271} = 0)$	$(p_{271} = 1)$
0	0	0.963	0.037
0	1	0.635	0.365
1	0	0.882	0.118
1	1	0.582	0.418

In contrast, consider the case illustrated in Table 4. The conditional probabilities of product p_{533} indicate that the purchase probability of p_{533} drop when the parent products, p_{170} and p_{65} , are included in the basket. This is a sign of weak associations of a child product with its parents indicating that it is not very likely that they are going to be included in the same shopping basket.

Table 4: The CPT for p_{533}

CPT: p_{533}			
p_{170}	p_{65}	$(p_{533} = 0)$	$(p_{533} = 1)$
0	0	0.956	0.044
0	1	0.981	0.019
1	0	0.988	0.012
1	1	0.986	0.014

A way of measuring the degree of change on the conditional probabilities of the child node depending on the states of its parents is to apply the distance formula for every possible pairs of the conditional probabilities and then average them.

$$d = \sum_i (p_i - q_i)^2 / \binom{n}{2}, \forall p \wedge q \quad (3)$$

In this formula, d indicates the average distance of the variable of interest that is a child node. Here, p and q stand for the conditional probabilities of this child node for the different states of its parents, i stands for the different states of the child node and n stands for the number of states of the set of parent nodes.

A high average distance is desired as an indication of the high association of the child node with its parents and can be used as a reference to select the products used in the final BN created. The average distances of p_{271} and p_{533} are calculated as 0.068854 and 0.0004 respectively. With p_{271} having a higher average distance than p_{533} , our prior conclusions are verified.

However, the selection of the products shouldn't be based on average distances solely since there may be products that are poorly associated with their parents but highly associated with their children. As an example, consider the case illustrated in Figure 2. Product p_{270} has p_{223} as its parent and from prior analysis we know that p_{270} has the product p_{271} as its child. Looking at the conditional probability table illustrated in Table 5, the average distance of p_{270} from its parent is calculated as 0.013482, which suggests that p_{270} should not be selected to the final BN to be created. However, from our prior findings we know that having p_{270} in the cart has a huge impact on the purchase probability of p_{271} . Not selecting a product like that would just deteriorate our findings.



Figure 2. Product p_{270} with p_{223} as its parent and p_{271} as its child

Table 5: The CPT for p_{270}

CPT: p_{270}		
p_{223}	$p_{270} = 0$	$p_{270} = 1$
0	0.957965	0.042034
1	0.841851	0.158149

Ideally, we should find the distance in conditional probabilities of a parent node with its children, sum it with the average distance of the same node with its parents and use the result obtained to select the products for the BN. Nevertheless, although the capture of this information is possible, it requires time-consuming calculations. For that reason, the selection process will be done following the heuristic described below.

Notice that, the average distance in the conditional probabilities obtained through BNs shows the associations of the child node with its parents. This is another way of saying that it shows the association of a parent node with its child, jointly with the child's other parents. Hence, the average distance in conditional probabilities can also be used as a reference for the level of association of a node with its child. Accordingly, as the first step of the heuristic, we will calculate the average distance of each variable's conditional probabilities.

Let d_j denote this average distance, where j denotes variable considered. Using d_j , a score called S_j will be calculated which is the sum of the distances of the variable of interest and its children. Thus,

$$S_j = d_j + (\sum_i d_{ij} / C_j) \quad (4)$$

where index i denotes the child node(s) of the variable j and C_j denotes the number of j 's children. S_j will be used as a reference to select the products that are going to be used in the final BN. Calculation of S_j for the product p_{89} is illustrated in Figure 3 below.

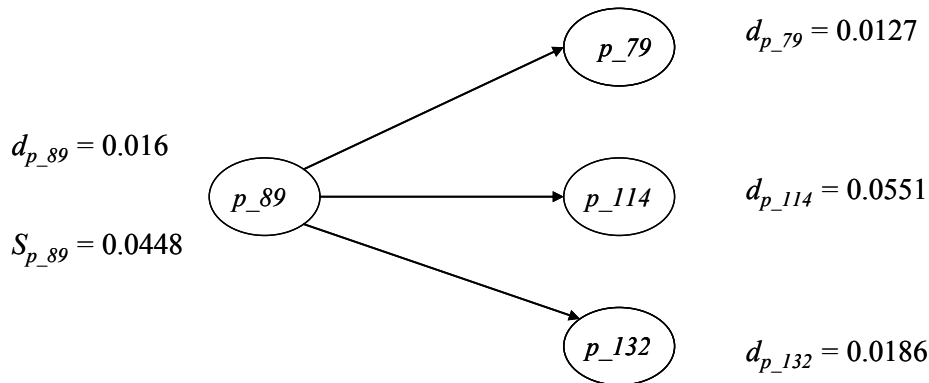


Figure 3. Calculation of S_j for the product p_{89}

The heuristic is as follows: The variable with the lowest S_j is the variable that has the lowest level of associations with the other variables. Hence, that variable will be excluded from the analysis. With the variables remaining, a new BN will be created and from the conditional

probability tables obtained through this new BN new S_j scores will be calculated. These steps, (exclusion of the variable with the lowest S_j score and creation of the new BN) will be repeated until we end up having as many variables remaining as the number of variables that we want to use in the final BN.

Using this heuristic we will be able to find the products that have high associations with each other, hence the products that are likely to be purchased on the same shopping trip. Also, with the help of this heuristic, we are able to detect the association between the products in the dataset, which are not present to a very high degree in the database and may be overlooked in the analysis of the aggregate data. On the other hand, the current problems involved with this heuristic are mostly because of some technical difficulties. We do not have a system to automate the procedure. We obtain the CPTs from the BNs created in WinMine. However, WinMine does not allow copying the CPTs, so after printing the tables and recreating them in Excel we find the average distances. Because of this problem, we chose the top 10 products with the highest S_j scores instead of discarding the variables with the lowest S_j scores one by one.

The final BN corresponding to the 10 products with the highest S_j scores is learned from data using the WinMine toolkit where the dataset is divided into a training and a test dataset. A train/test split is performed with 7897 cases as for the training and 3385 cases for test which correspond to 11282 baskets in total. All of the variables are used as input-output variables (both predicted and used to predict). Accordingly, as given in Table 1, the BN created using the heuristic proposed results in a log score of -0.124822 , meaning on average, the probability that each variable assigns to the given value in the test case, given the value of other variables, is 91.71%. The lift over marginal log score is 0.229405, which shows with our model using the information of the products in cart we obtain 13.5% improvement in the average probability of correct prediction compared to the marginal model⁷. For comparison purposes we also created a BN from the dataset of the most purchased top 10 products and provided the corresponding log-scores in Table 1 above. As seen from Table 1, if we don't use the heuristic and use the top 10 products with the highest frequencies instead, the improvement obtained would be only 3.613%. Therefore, we conclude that the performance of the proposed heuristic is quite good.

⁷ The average probability of correct prediction for the marginal model is 78.2288 % in this case.

The BN created using the proposed heuristic with top 10 products with the highest S_j scores is illustrated in Figure 4 below. In the following section the performance of the proposed heuristic is evaluated by applying it to modified versions of the original data set.

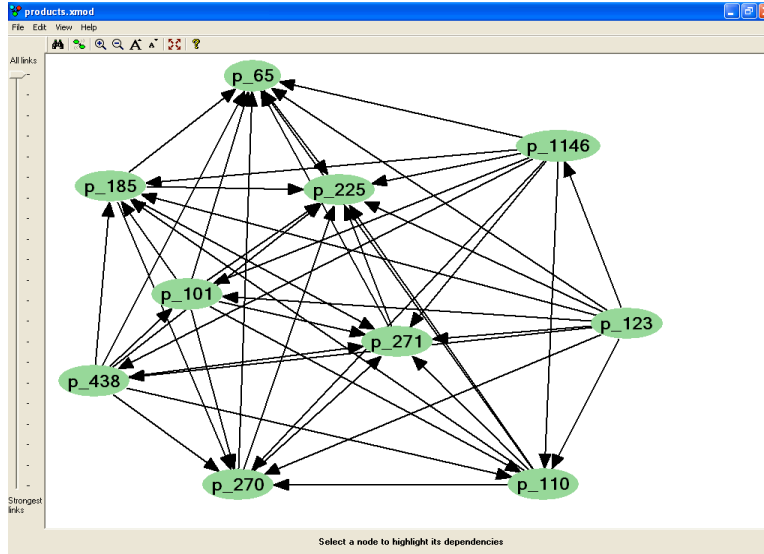


Figure 4. The BN created 10 products with the highest S_j scores

4.2 Validation of the Proposed Heuristic

In subsection 4.1 the proposed heuristic and its' results are presented. As improvement is one of the four dimensions of a heuristic (the other three being uncertainty of outcome, basis in incomplete knowledge and guidance of decision making) it is a certain necessity for a newly proposed heuristic that its performance is evaluated (Romanycia *et al.*, 1985). According to the results, the BN model created using the proposed heuristic obtained a 91.71 % prediction rate in the test data which is used as the holdout data and constitutes to 30% of the whole data set. The lift over marginal values suggest that the prediction accuracy obtained compared to the marginal model corresponds to 13.4829% improvement. These results are a nice indication about the sound performance of the proposed heuristic especially apparent when the performance details of the previous BNs⁸ are analyzed. Complete details can be found in Table 1 given in the previous section.

⁸ BNs created without the using the heuristic

Since there is not much literature for learning BNs from sparse data sets in the context of its performance for a recommendation system, and as this heuristic may be considered as a new approach in this area, in this subsection we apply the proposed heuristic again to validate our findings. For that purpose the original data set used in the previous section is converted into 4 new data sets where each time a different 25% of the data lines are deleted from the original data. Repeating the pre-described steps of the proposed heuristic, we first identify the most purchased 30 products in each of these new data sets and then create BNs using only the data for the most purchased 30 products in the data sets⁹. The logscore and lift over marginal values of these Bayesian networks along with their corresponding prediction and improvement rates are given in Table 5, 7 and 9 consequently. The average improvement rate obtained using these new data sets for the most purchased 30 products is 1.1647 % which is not much different than the improvement obtained with the original model, 1.1813 %.

Table 5: Log-score and lift over marginal values for all the data sets

	Most frequent 30		Most frequent 6th -30th		Heuristic 10	
	Log-score	Lift over marginal	Log-score	Lift over marginal	Log-score	Lift over marginal
Data 1	-0.174938	0.018499	-0.195047	0.011635	-0.155544	0.209779
Data 2	-0.185513	0.019088	-0.20844	0.010573	-0.153892	0.216033
Data 3	-0.185867	0.019578	-0.209756	0.010401	-0.167253	0.203689
Data 4	-0.189356	0.019697	-0.208455	0.010124	-0.150739	0.213405
Original data	-0.186378	0.0195231	-0.207396	0.0114608	-0.124822	0.229405

As the next step, considering the low improvement rates of the provided models, we investigate the frequencies and proportional frequencies of the seven most purchased products to its following one in each of the four data sets. The figures given in Table 6 below portray a similar picture as the one for the original data set. In all of the four data sets we see a huge increase in the proportional frequency of the fifth most purchased product whereas it drops again by the sixth product. Notice that the top five most purchased products are the same for all the

⁹ While some products are common for each of the most purchased 30 products lists (mostly the most purchased top five), the variety of the products included in the data sets are different for all of the four data sets.

newly created data sets. This is an expected result since as illustrated in Section 3.2 there is high discrepancy between the purchase frequencies of the most purchased five products and the rest of the data set. Anticipating that these five products are bought on a daily basis we exclude them from the analysis and create BNs with the remaining 25 products for further analysis. However, for these networks the average improvement rate even drops to 0.64 %, similar to the case observed in the analysis of the original data.

Table 6: The frequencies and proportional frequencies of the seven most purchased products for all data sets

Data Set 1	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{89}
Frequency	22047	18519	7093	6857	6537	2055	1649
Proportion	1.191	2.611	1.034	1.049	3.181	1.246	1.065
Data Set 2	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{89}
Frequency	21737	17874	6799	6765	6575	1888	1635
Proportion	1.216	2.629	1.005	1.029	3.483	1.155	1.064
Data Set 3	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{89}
Frequency	21939	17673	7075	6746	6717	1861	1764
Proportion	1.241	2.498	1.049	1.004	3.609	1.055	1.236
Data Set 4	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{170}
Frequency	21759	18182	10554	6791	6743	1858	1492
Proportion	1.197	1.723	1.554	1.007	3.629	1.245	1.032
Original Data	p_{39}	p_{48}	p_{41}	p_{38}	p_{32}	p_{65}	p_{89}
Frequency	29288	24201	10554	9100	8897	2562	2142
Proportion	1.210	2.293	1.160	1.023	3.473	1.196	1.088

As the last step of the heuristic using the information obtained from the $cpts^{10}$ we calculate the S_j scores for all the variables. Choosing the top 10 products with the highest S_j scores in each data set we create the final BNs and compare their average performance to the BN obtained from the original data. The list of the 10 products for each data set is given in Table 8 below. Notice that while there are common products in each of the five data sets (four new and one original) there is not a single product which is common in all of the data sets showing a mixed product variety for data sets. As detailed in Tables 5, 7 and 9 the results indicate that by applying the

¹⁰ The $cpts$ are obtained from the BNs created with the 6th-30th most purchased products

proposed heuristic to new data sets, we obtained BNs where the average rate of prediction is 89.6986 % and the average improvement obtained with the provided model is 12.1891%. Comparing these results to the performance of the original data set we only see a small difference. For the original data set the prediction rate was 91.7177% and improvement rate was 13.4829 %. That the performance of the original network is better than the new data sets is an expected result since the original BN bases its analysis on complete data set whereas the other four refer to only 75% of the data set. Remember, the new data sets are created by deleting a different 25% each time. Hence we conduct the same analysis for each of the four data sets and then obtain the average results. With deletion of a smaller portion of data and a greater number of repetitions of the analysis consequently, the performance of the new data sets would approach to the performance of the original data set in a great deal.

Table 7: Prediction rate with the provided model

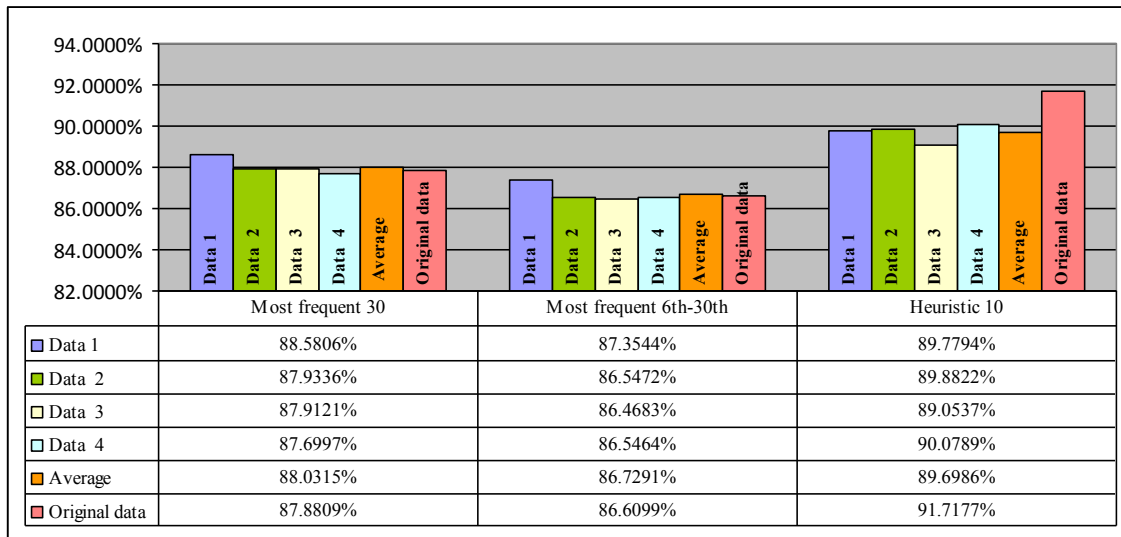
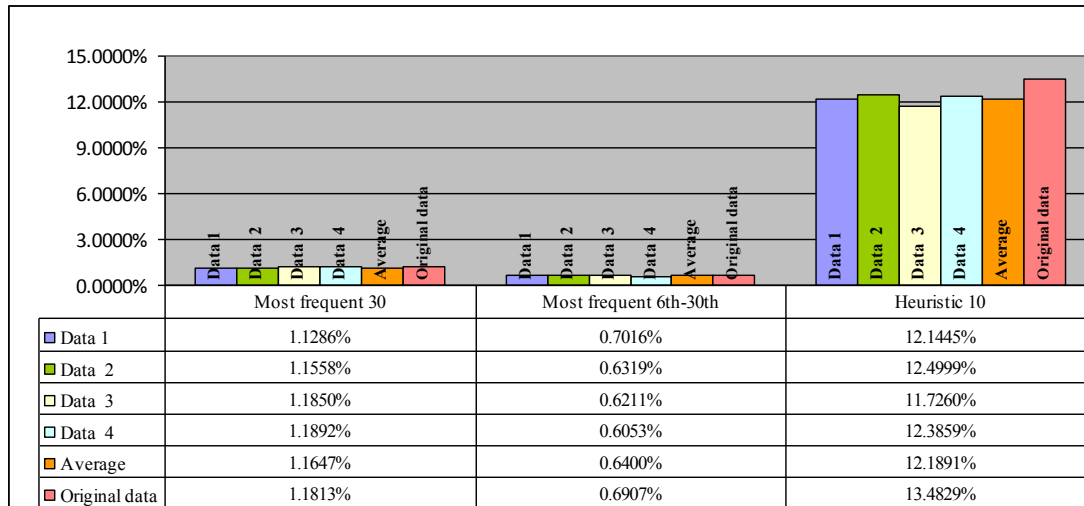


Table 8: The list of the products selected using the proposed heuristic

Data Set 1	Data Set 2	Data Set 3	Data Set 4	Original Data
<i>p</i> ₆₅	<i>p</i> ₇₉	<i>p</i> ₆₀	<i>p</i> ₃₆	<i>p</i> ₆₅
<i>p</i> ₈₉	<i>p</i> ₈₉	<i>p</i> ₆₅	<i>p</i> ₆₀	<i>p</i> ₁₀₁
<i>p</i> ₁₀₁	<i>p</i> ₁₀₁	<i>p</i> ₈₉	<i>p</i> ₆₅	<i>p</i> ₁₁₀
<i>p</i> ₁₈₅	<i>p</i> ₂₂₅	<i>p</i> ₁₀₁	<i>p</i> ₁₁₀	<i>p</i> ₁₂₃
<i>p</i> ₂₂₅	<i>p</i> ₂₇₀	<i>p</i> ₂₂₅	<i>p</i> ₁₇₀	<i>p</i> ₁₈₅
<i>p</i> ₂₇₀	<i>p</i> ₂₇₁	<i>p</i> ₂₃₇	<i>p</i> ₂₂₅	<i>p</i> ₂₂₅

p_{271}	p_{310}	p_{270}	p_{271}	p_{270}
p_{310}	p_{438}	p_{271}	p_{310}	p_{271}
p_{438}	p_{475}	p_{604}	p_{1327}	p_{438}
p_{475}	p_{2238}	p_{2238}	p_{2238}	p_{1146}

Table 9: Improvement rates obtained by the provided model compared to the marginal model



5 A Case Study

In the previous section, we described a new heuristic for learning BNs from limited datasets. In this section, using the probability tables obtained by WinMine, the same BN will be constructed in *Hugin*, a commercial BN implementation. The advantage of using *Hugin* is that it allows entering evidence to the BN created. In *Hugin*, using the ‘sum propagation’ method the posterior probabilities of the variables can be computed according to the evidence received. Additionally, using the ‘max normal’ propagation method the states belonging to the most probable configuration can be found. The state of node with the most probable configuration is given in units of 100. The values for all other states are the relative probabilities given that the other nodes are in the most probable configuration.

The conditional probability table used for the product p_{101} is illustrated in Table 10. Notice that there are two initial states for each product, ‘0’ and ‘1’, representing that the product is ‘not in cart’ and ‘in cart’, respectively. The BN created using *Hugin* and the marginal probabilities of the products obtained (using the ‘sum propagation’ method) are given in Figure 5 below.

Table 10. The conditional probability table of p_{101} given its parents

CPT: p_{101}				
p_{123}	p_{438}	p_{1146}	$p_{101} = 0$	$p_{101} = 1$
0	0	0	0.860	0.140
0	0	1	0.975	0.025
0	1	0	0.985	0.015
0	1	1	0.888	0.112
1	0	0	0.968	0.032
1	0	1	0.857	0.143
1	1	0	0.956	0.044
1	1	1	0.5	0.5

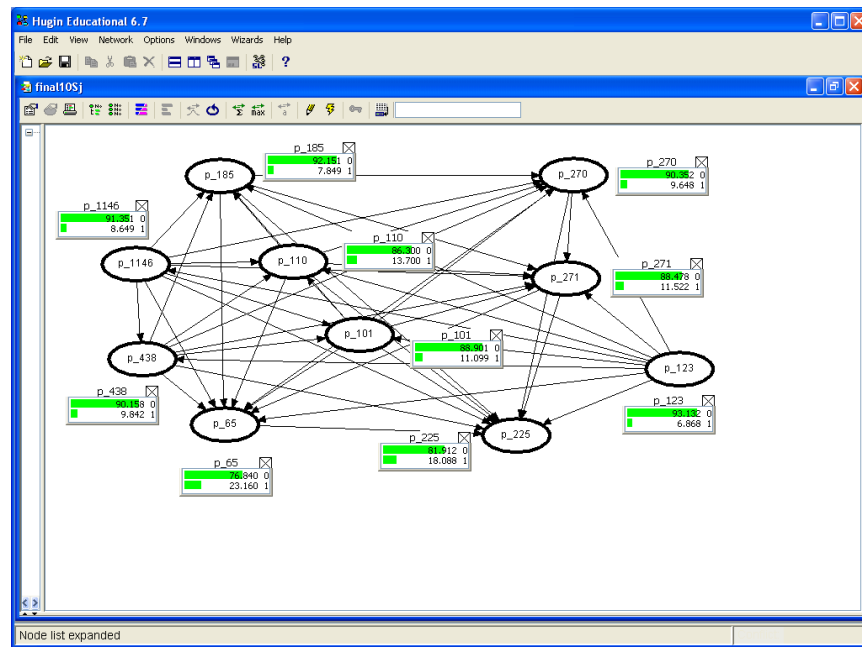


Figure 5. The final BN and the marginal probabilities of the variables obtained in Hugin

Table 11. Marginal probabilities and most likely states of the products in the BN

	<i>Marginal for '0'</i>	<i>Marginal for '1'</i>	<i>Most likely state</i>
p_{65}	76.840 %	23.160 %	1
p_{101}	88.901 %	11.099 %	0
p_{110}	86.3 %	13.7 %	0
p_{123}	93.132 %	6.868 %	0
p_{185}	92.151 %	7.849 %	0
p_{225}	81.912 %	18.088 %	0
p_{270}	90.352 %	9.648 %	0
p_{271}	88.478 %	11.522 %	0
p_{438}	90.158 %	9.842 %	0
p_{1146}	91.351 %	8.649 %	0

Looking to the most likely states and the marginal probabilities of the products which are given in Table 11, we see that for the state ‘1’ the product p_{65} has the highest marginal probability as 23.16% and with 6.868% the product p_{123} has the lowest marginal probability. Other than the product p_{65} , the most likely state for all the products is ‘0’.

Consider a scenario where two promotions are going on inside a grocery store, a promotion for the product p_{225} and a promotion for the product p_{65} . The task of the manager is to decide to which customer to advertise which one of these two products. Based on the marginal probabilities the product p_{65} should be chosen since with 23.160 % it has a higher marginal probability for the state ‘1’ compared to its counterpart p_{225} . But being able to observe what the customer is placing in her/his cart, we may change our recommendation.

Suppose we observe a customer who places the products p_{110} and p_{123} in his basket. Accordingly, we decide to recommend p_{225} to that particular customer, since after observing what is placed inside the cart the posterior marginal for p_{225} increases to 36.4%, whereas for p_{65} it is 23.2 %. As the second case, suppose we observe a customer with three different products placed inside her cart, the products p_{101} , p_{271} and p_{1146} . With these products inside the cart the posterior marginal probability for the product p_{65} increases to 56.170 %. With p_{65} , having the higher probability to be purchased by the customer, p_{65} will be recommended. The BN after observing the products in cart concerning the second customer is given in Figure 6 below. The details of the case and the associated probabilities are summarized in Table 12 below.

Table 12. The posterior probabilities and most likely states of p_{225} and p_{65} for the state ‘1’

<i>Information</i>	$p_{225} = 1$		$p_{65} = 1$	
	<i>Marginal</i>	<i>Most likely state</i>	<i>Marginal</i>	<i>Most likely state</i>
<i>Prior</i>	18.1 %	0	23.2 %	1
<i>Cart 1</i> ($p_{110} = 1, p_{123} = 1$)	36.4 %	0	23.2 %	0
<i>Cart 2</i> ($p_{101} = 1, p_{271} = 1, p_{1146} = 1$)	45.9 %	0	56.2 %	1

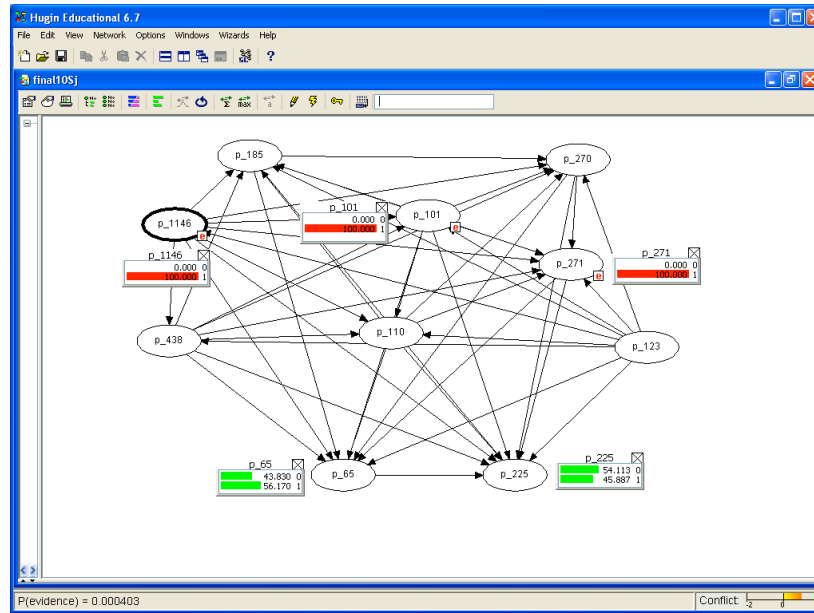


Figure 6. The updated BN concerning the second customer with 3 products in cart

6 Summary and Conclusions

The main contribution of this paper is a new heuristic for learning Bayesian networks from small datasets. We demonstrate the use of this heuristic for a grocery store model with RFID systems where Bayesian networks are used as a tool for real time recommendation. Though very suited as a dynamic decision making tool, Bayesian networks are difficult to learn from small datasets. In the literature, there are many BN learning algorithms and heuristics for sparse datasets. Most of these approaches are only concerned with the general quality of the Bayesian network and do not relate to the usefulness of the learnt model in the context of an application. However, for a dynamic recommendation system (such as the futuristic grocery store example in this paper), the usefulness of the inferences is a big concern. Accordingly, the heuristic that we describe in this paper should be classified as a heuristic for the determination of the variables to be included in the BN to be learned. The purpose is to reveal the information hidden inside sparse datasets to form useful recommendations.

Using the methods of this new heuristic we were able to expose the information contained in a limited dataset to find the products that have high associations with each other, hence the products that are likely to be purchased on the same shopping trip. In addition to that, this new heuristic may detect the association between the variables in the dataset, which are not present to

a very high degree in the database and may be overlooked because of that in the analysis of the aggregate data. As the corresponding log-scores suggest, using the proposed heuristic better performing networks may be achieved also from sparse datasets, such as grocery. The BN created using the heuristic proposed does have 91.71% average probability of correct prediction which is 13.48% better than its marginal model. We evaluated the proposed heuristic and validated its performance by applying it to four new data sets which are created by deleting a different 25% of the original data each time. As a future research topic the proposed heuristic may be programmed in a general purpose API using Hugin or Genie. This way we may have a more user-friendly environment and be able to automate the procedure described.

In e-commerce retailers are capable of making real-time recommendations to customers and hence influence the decision making process of the customer at the same time as the customer is shopping. The model we used in this paper, first introduced in Cinicioglu *et al.* (2007), transfers the methods of e-commerce to grocery stores that will have a huge impact for the success of the promotions and acceleration of demand. In that instance, it should be mentioned that the majority of the visitors of a grocery store are real customers and not just browsers as it is the case with e-commerce. Also, although we refer to the model as futuristic, it is our belief that this grocery store model will become feasible in the near future, with Metro AG testing its future store initiative and the cost of RFID decreasing to 5 cents a tag (SC Digest, 2009). Regarding the ongoing debate about the privacy implications of the RFID tags, notice that the data used for our recommendation system is no different than the checkout data of the customers that is already collected by the most of the grocery stores via loyalty cards. As a future research topic, the longitudinal information about these users can be used to improve the effectiveness of this system.

Acknowledgement

This research was partly supported by Istanbul University research fund project number 6858. We are grateful for three anonymous reviewers of AnOR for comments and suggestions for improvements.

References

- Breese, J.S., Heckerman, D., Kadie, C., (1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering, In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI, Morgan Kaufmann Publisher
- Brijs T., Swinnen G., Vanhoof K., and Wets G. (1999), The Use of Association Rules for Product Assortment Decisions: A Case Study, in: *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, San Diego (USA), August 15-18, 254–260. ISBN: 1-58113-143-7.
- Cavoukian, A. (2004), Tag, You're It: Privacy Implications of Radio Frequency Identification (RFID) Technology, *Information and Privacy Commissioner*.
- Cinicoglu, E.N., Shenoy, P.P, Kocabasoglu, C. (2007), Use of Radio Frequency Identification for Targeted Advertising: A Collaborative Filtering Approach Using Bayesian Networks, in K. Mellouli (ed.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Lecture Notes in Artificial Intelligence*, **4724**, 889 –900, Springer-Verlag, Berlin.
- Cozman, F. G., (2000), Credal networks, *Artificial Intelligence*, **120**(2), 199 – 233
- Cui, G., Wong, M. L., Zhang, G., (2010). Bayesian Variable Selection for Binary Response Models and Direct Marketing Forecasting, *Expert Systems with Applications*, **37**, 7656 –7662
- Finkenzeller, K. (1999). RFID Handbook Radio-Frequency Identification and Applications, *John Wiley & Son*.
- Friedman, N., Goldszmidt, M., Heckerman, D. & Russell, S. (1997). Challenge : What is the impact of Bayesian networks on learning?, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (NIL-97)*, 10–15
- Friedman, N., Nachman, L. and Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The “sparse candidate” algorithm. *Proc. Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI' 99)*, 196–205.
- Goldenberg, A., Moore, A., (2004). Tractable Learning of Large Bayes Net Structures from Sparse Data. *Proceedings of 21st International Conference on Machine Learning*
- Gu, Q., Cai, Zhihua, Zhu, L., Huang, B., Data Mining on Imbalanced Data Sets, *Proc. International Conference on Advanced Computer Theory and Engineering*, pp 1020–1024
- Heckerman, D., Chickering, D. M., Meek, C., (2000) Rounthwaite, R., Kadie, C., Dependency Networks for Inference, Collaborative Filtering, and Data Visualization, *Journal of Machine Learning Research*, **1**, 49–75
- Hicks, P. (1999), RFID and the Book Trade, *Publishing Research Quarterly*, **15**(2), 21–23
- Jaroszewicz, S., Simovici, D., A., (2004). Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge, *Proceedings of the 2004ACM SIGKDD International Conference on Knowledge Discovery and data Mining*, Seattle, WA, USA, pp. 178 – 186
- Liu, B., Zhao, K., Benkler, J., and Xiao, W. (2006). Rule Interestingness Analysis Using OLAP Operations, *Proc. ACM KDD*, 297–306

- Liu, F., Tian, F., Zhu, Q., (2007), An Improved Greedy Bayesian Network Learning Algorithm on Limited Data, in Marques de Sá et al. (ed.), ICANN 2007, *Lecture Notes in Computer Science*, **4668**, 49–57, Springer Berlin/Heidelberg.
- Madsen, A., Lang, M., Kjaerulff, U. and Jensen, F., (2004). The Hugin Tool for Learning Bayesian Networks. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Springer Berlin/Heidelberg, 2711, 594–605
- Mild, A. (2003), An Improved Collaborative Filtering Approach for Predicting Cross-category Purchases Based on Binary Market Basket Data, *Journal of Retailing and Consumer Services*, 10, 123–133
- Oniśko, A., Druzdzal, M. J., Wasyluk, H., (2001), Learning Bayesian Network Parameters: Application of Noisy-OR Gates, *International Journal of Approximate Reasoning*, **27**, 165-182
- Pine II, B.J. (1993). Mass Customization, *Harvard Business School Press*, Boston. Massachusetts
- Pine II, B.J. and Gilmore, J.H. (1999). The Experience Economy, *Harvard Business School Press*, Boston. Massachusetts
- Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., Riedl, J. (1994): GroupLens: An Open Architecture for Collaborative Filtering of Netnews, In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 175–186
- Resnick, P. and Varian, H. (1997). Recommender Systems, *Communications of the ACM*, **40**(3), 56–58
- SC Digest Editorial Staff, (January, 2009) The Five Cent RFID Tag is Here, <http://www.scdigest.com/assets/newsviews/09-01-27-2.pdf>
- Romanycia, M. H., Pelletier, F. J., (1985) What is a Heuristic?, *Computational Intelligence*, **1**, pp. 57-58
- Scuderi, M., Clifton, K. (2005). Bayesian Approaches to Learning from Data: Using NHTS Data for the Analysis of Land Use and Travel Behavior, Bureau of Transportation Statistics, US Department of Transportation, Washington, DC.
- Whitaker, J., Mithas, S., Krishnan, M. S. (2007), A Field Study of RFID Deployment and Return Expectations, *Production and Operations Management*, **16** (5), 599–612
- Yu, K., Schwaighofer, A. Tresp, V. Xiaowei, X., Kriegel, H.P. (2004) Probabilistic Memory-based Collaborative Filtering, *IEEE Transaction on Knowledge and Data Engineering* **15**(1), 56–69