5-2018

Realistic Market Basket Data Simulation for Retail Marketing

Elizabeth N. Nichols
The University of Tennessee - Knoxville, enichol6@vols.utk.edu

Realistic Market Basket Data Simulation for Retail Marketing

Elizabeth Nichols
Business Analytics & Statistics – Haslam College of Business
The University of Tennessee - Knoxville

Advisor: Wenjun Zhou
Business Analytics & Statistics – Haslam College of Business
The University of Tennessee – Knoxville

ABSTRACT

The objective of this project is to assess the effectiveness of three simulators for potential use in sales prediction and business scenario generation. We use a large point-of-sales dataset chain where each transaction is considered a 'basket' consisting of items purchased.

First, we test the performance of two existing transaction data simulators; one method that assumes independence among items and the other which considers co-purchase correlation. We split the data into two portions: a randomly chosen week's data as the holdout set for performance assessment, and the remaining portion as the training set for identifying data patterns for simulation modeling. We then simulate one week's worth of baskets. The simulated baskets are compared to the actual baskets in the holdout to see how close they resemble the actual. Our hypothesis is the correlated version will provide a more realistic simulated sample than the independent approach.

Based on the findings, we develop our own simulation model to outperform the two above. We utilize insights and shortcomings seen in models' evaluations to guide our choices in aspects to add and withhold. We evaluate the data utilizing the same training and holdout data portions as before for equal comparison and accurate conclusions. The consideration of the training data's characteristics was the main element we chose to include to enhance the existing methods. We find this new simulator performs better than the two established methods in the three distributions considered in this study.

This improved simulator can help better exemplify shopper behaviors as seen in the actual retail data. As a result, retailers can more accurately generate benchmark datasets for evaluation of developed algorithms.

INTRODUCTION

With the rise of technology use in industry combined with the relative ease of customer data collection, retailors today have massive collections of transactions data, which can be analyzed for business insights and decision-making. Big data such as these have been identified with three fundamental characteristics: *volume*, the amount of data collected; *velocity*, how close to real-time the data is being collected; and *variety*, the various formats in which the data is stored (Laney). Businesses individually determine how to collect their data with respect to these three characteristics based on their needs and collection capabilities. The range of possible analytics studies is then determined from the characteristics of the collected dataset. The importance of big data lies more in how businesses utilize it than the quantity available.

In retail, businesses desire to utilize the data they collect to accurately represent what is happening in their establishments. The goal of this study is to assess the accuracy of three sales simulators and their leverage for prediction and business experimentation. Retailors can improve service and business performance by better understanding the purchasing habits of their customers in various situations and these simulations can be used as a benchmark for comparison. We simulate data based on the retailer's original data and evaluate the accuracy by comparing the item frequency, basket size, and average basket value distributions to the original data's distributions. By evaluating how realistic the three simulators are in these areas, we can choose which one performs most accurately for recommendation of business use.

In order to complete the study accurately, we will primarily work with a simplified dataset containing just the transaction and item IDs. The data will be split into a holdout sample of 1 week's transactions with the other weeks' transactions left out as the training data. Using the training data, three simulators will attempt to model the holdout data. The simulations will be compared to the actual data utilizing the measure of KL Divergence to determine how close the simulated transactions resemble what they are supposed to be modeling. Based on these comparisons we will be able to accurately determine the best model by which to simulate retail data.

Our study compares three different methods to showcase different aspects of simulation versus reality similarity. Two of these methods are from an existing package in R. One considers independence among items and the other considers co-purchase correlation among items. We propose a third method to improve upon the two methods. The newly developed simulator proves to be more realistic in mimicking the item frequency, basket size, and basket value distributions. The improved simulator can be used for generating benchmark datasets for evaluating new algorithms, and it may provide basis for advanced tools for retailors to create and meet realistic financial goals, to experiment with new scenarios embracing low cost, and to optimize marketing practices in general.

The rest of this paper is organized as follows. We first provide a brief literature review. Then an overview of the data utilized for this report. Next move into the methods section consisting of the study design followed by an explanation of the existing simulators, our simulator, and the measure of comparison. Our results for the item frequency, basket size, and basket value distribution comparisons follow. The report is ended with a section of conclusions and areas of further study. The references and R code used for the research are provided for clarification.

LITERATURE REVIEW

It is generally recognized that the data collected by retailors is large and complex, but rich in information for the company. The data for this study contains over two million records for just two year of data collection. Quick algorithms and methods of evaluation are needed to extract that collected information for productive use (Agrawal & Srikant). A big area of interest in utilizing retail data is mining patterns from the data to improve some area of the business. Examples include finding items frequently bought together, determining common purchases of customers from different demographics, and so on. Without fast and effective methods however, these datasets are too large to evaluate with simpler approaches and the data ends up sitting with unused potential or thrown out because the business believes it cannot be used. We take necessary steps in our data processing to ensure only the needed pieces information are present for simulation to cut down on running time.

There are three major issues that have to be overcome when working with market basket data in order to have the most success data mining from it. The first looks at classification; figuring out how to partition the data to match the task at hand (Agrawal). Since this research includes model building it is important that the original data is split into a training dataset for use with simulators and a holdout dataset for simulation comparison. The second is associations between items and determining which if any are appropriate to note and consider (Agrawal). This will be discussed in more detail later, but between the three simulators we consider, the one that considers correlations between items performs the worse. In our simulator development it was decided that all potential item associations will be recognized as null. The last is sequences among data points, especially time, where the order can effect the analyses or serve as an extra point of clarification (Agrawal). Time is a noted variable of our original data, but one that is not considered in the simulations because it is assumed that data variation does not change significantly from week to week. This could be a drawback to our conclusions if this assumption is not accurate.

The two established simulators utilized in this study come from random.transactions function within the "Arules" R package. While both use the same function, there are two separate methods available for input, giving two unique simulation methods. The function generates a simulated database of number of items by number of transactions called a "transaction class" in R terminology.

The first method is "Independent", labeled as such because it considers all items to be independent from one another and picks transaction size based off a Poisson distribution with a center of three (Hahsler, Hornik, & Reutterer). Items are then all given the same probability of being chosen and are picked randomly based from the uniform probabilities. (Hahsler, Hornik, & Reutterer).

The second method is "Agrawal", which considers correlations among items with six additional parameters users can specify when creating transactions (Agrawal & Srikant). This method also utilizes a Poisson distribution for the patterns between the items specified with a user entered mean (default = 4), each pattern from this distribution is assigned a weight from an exponential distribution with a mean of one (Agrawal & Srikant). This step is done within a partner function "random.patterns" that can be used to override the default patterns parameters for the random.transactions calculation. From the specified

patterns the length of transactions is pulled from a Poisson distribution with a mean of the average length of patterns (Agrawal & Srikant). Then, patterns are randomly chosen for each individual transaction using the pattern weights created with random.patterns functions or determined by the default values until it matches the basket size (Agrawal & Srikant). This is repeated until the desired number of transactions are created.

Simulations of market basket data, like what can be created with the two simulators described above can be used for a plethora of purposes depending on the individual needs of businesses doing analysis. One example is looking at a retailer's supply chain combined with simple data collection to be able to help predict how the retailer can price their items more competitively (Besanko, Dube, Gupta). From data collected about the purchases and promotions of a certain product paired with the assumptions of supply chain costs, the business is able to break the customer population into three segments based on their sensitivity to price and sales displays and determine the price elasticities for each group (Besanko, Dube, Gupta). This data sounds very similar to *The Complete Journey* dataset we use in this study containing information about item sales and the promotions associated with them. This information, all mined from data, allows the company to have accurate simulations of how many customers will purchase the item at different prices, which in turn allows them to account for how many units of the item they need to have stocked in store. This same process can be repeated for other commonly purchases items in the store to help save on costs incurred by poor supply chain management as well as help know how to market items more efficiently to reduce an oversupply without a major impact on profit. Instead of simulating one item, however we aim to simulate the entire range of transactions over a week, saving time on the process of determining what the entire store can expect with transactions to plan accordingly.

Another example is looking at the brand preference of customers within retail by simulating their transactions based on their past history to look for patterns and have more accurate customers segmentation (Russell & Kamakura). The example study kept their model very simple, not considering any outside variables that may affect purchases, such as price, and evaluated customers' brand purchases of paper products (Russell & Kamakura). This matches the method in which we base our simulations in order to create a broad simulator that can work for any retail data. They use a Poisson distribution for the base of their model for tractability, but add on the factors of customer intensity, the fact that different brands do not have to be forced into substitutes, and household category to account for the biggest factors they observed in their collected data (Russell & Kamakura). Again this follows our method of starting with a simple distribution and expanding on it with variables that can be altered with different datasets to increase accuracy. Both are expanded to account for the shortcomings of variables that are originally not accounted for. From this data, they were able to better simulate a customer's long-term purchase behavior as well as more accurately segment customers into groups with similar shopping habits by brand preference (Russell & Kamakura). We expect our improved model to yield the same results of increased accuracy from the simpler model we used originally.

DATA DESCRIPTION

A point-of-sales dataset, called *The Complete Journey*, was obtained Dunnhumby, a global customer analytics company for retailers and brands. This dataset contains

transactions from a group of two thousand, five hundred households over two years, all of whom are frequent shoppers at a specific retailer.

The main data file, *transaction_data.csv*, is formatted where each row corresponds to a transaction-item pair. The Basket_ID labels the unique transactions and all the retailer's unique items have individual UPC codes, labeled as the Product_ID. The data is rich with multiple explanatory variables that allow for extensive analysis of transaction contents. Unused variables include the unique household key for each of the household shoppers, day on which the transaction was made, store ID of where the transaction was completed, retail discount of the item bought, time the transaction occurred, and discount from using coupons. Each variable is listed in a column of the dataset. In total, this dataset has 2,595,732 records consisting of 276,484 unique transactions and involving 92,339 unique items.

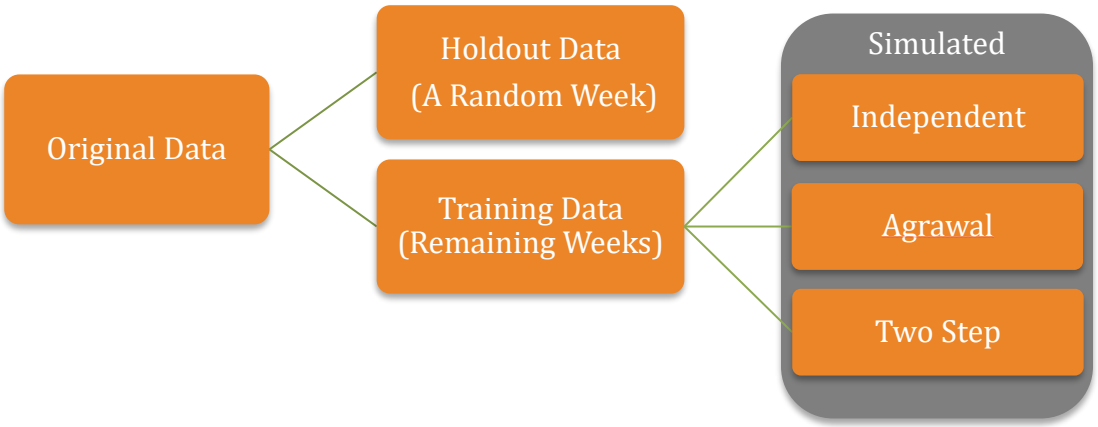
There are a few other linkable data files that provide details about promotions, product categories, household demographics, etc. However, these details are ignored in our study. As a result, outcomes are focused on the simulation as whole instead of data specifics, which is preferable for the goals of this study. We have also ignored the promotions and discount information on price in the main data file. By ignoring the promotions information, our implicit assumption is that people make purchases based on what they need, instead of being influenced by marketing efforts.

METHODS

Study Design

The original dataset had to be processed slightly before any examination of the simulators could be done. A copy of the data was reformatted with rows of unique Basket_IDs followed by the list of all Product_IDs purchased in the transaction. The original data was split into a list by Basket_ID with all the corresponding Product_IDs listed sequentially afterward. This list was then converted into the transaction format described in the R Arules package. The simulators will output the data in this same format, easing the comparison analysis seen later in the research. All other variables from the original dataset were withheld from this data copy to match the simplicity of the simulated output.

A Prices dataset was also created from the original dataset to allow us the ability to compare the accuracy of basket values between simulators. Since coupons and promotions were listed in separate categories, we decided to assume that the price of individual items would not vary much if at all from week to week. The sales value variable in the original dataset was divided by the quantity variable to create a "Prices" column of each individual item listed in the rows. Any item with a price of NaN or Infinity as a result of dividing by a quantity of 0 was then changed to $0.01. Since a penny's value in the United States is extremely low it felt this would not greatly impact the results utilizing this calculated variable. The Prices variable was then aggregated by PRODUCT_ID to create a new reference dataset only consisting of each individual item and it's mean price over the 102 weeks of data.



The week in which the transaction occurred, 1 to 102, is listed in the original data, allowing the data to be split into holdout and training samples by week. In model building and comparison this is vital because without it over-fitting, where a model is only accurate for the data on which it was created, is likely to occur. For simulation here the aim is to create and compare a general structure that can be used across the retail industry, making the split necessary. Additionally, a benefit of data splitting is the ability for the analysis and calculations to be performed at a much faster rate because of the reduced file size of the data.

A random number generator was used to select one week's transactions as the holdout data, for this study week 12 was randomly selected for use as the holdout data. From the data converted to transaction format a partition of the transactions was made to form two separate datasets, one with week 12's transactions and the other with 1-11 and 13-102's weeks of transactions. The first dataset is our holdout and the latter is the training data. The partition was made by arranging the list of Basket_IDs from the original data in the same order as the data in the transactions format. The week in which the transaction occurred was then matched to the Basket_IDs in a new dataset. Utilizing the Basket_ID/Week_No data, a subset of the rows with a Week_No equal to 12 was created. All of the transaction IDs observed in the data subset were pulled from the list created in the first step to create a vector of "selected transactions". This vector told what transactions needed to be pulled from the data in the transaction format to create the holdout data in the transactions format. The transaction IDs that were not selected and removed to create the holdout data were left creating by default the training data in the transaction format.

Utilizing the training data, three different simulators were run to try and replicate the holdout data. The first two, Independent and Agrawal are established methods and the Two Step is a method developed to improve upon the drawbacks noticed in the Independent method. With these three separate simulations, comparisons were made to the holdout with the measure of KL Divergence. Since the KL Divergence measures the difference between proportions, comparisons between the datasets' item frequencies, basket sizes, and basket values were made. This also allowed us to have a more comprehensive view of the strengths and weaknesses between the simulations.

Established Transactions Simulations

The simulated data was created using the random.transactions function in the R ARules package. Both the Independent and Agrawal methods were utilized to compare which simulation comes closer to replicating the actual holdout data. Example R code can be seen in the appendix of this report for a closer examination of how the code is formatted.

The Independent method treats all items as autonomous and baskets are chosen randomly from the list of items with uniform probability (Hahsler, Hornik, Reutterer). We hypothesize this method will not be very accurate because of dependence among items and the fact that certain items have a higher probability of being bought than others.

The Agrawal follows a Poisson distribution of patterns where the mean is specified by the user as well as the correlation between items (Agrawal, Srikant). For each pattern in the distribution, a weight is generated by drawing from an exponential distribution with a mean of 1 (Agrawal, Srikant). In order to see which correlation value would come closest to matching the actual value, correlations of -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, -0.05, -0.01, -0.001, -0.0001, 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9 were used with a mean of 2 and then 4 for patterns, creating the multiple simulated datasets.

Our Transactions Simulation

From an analysis of the strength of item pair co-purchases it was noted that for the original data there were practically no correlated strong pairs of items. The present item pairs were extremely weak in terms of correlation, so much so that the bottom threshold had to be continually lowered in order to get any to appear. Based on these results the conclusion was developed that retail datasets with high amounts of unique items, simply have too much variety in what customers can purchase to have patterns worth considering in simulation. This rules out Agrawal as an accurate method on which to develop a new simulation model.

The Independent model however, assumes a uniform distribution, which is not what is observed for the actual Week 12 data distribution. The information we felt was missing to improve the accuracy were the probability distributions seen in the training data, which we hypothesize will be reflected in the holdout data. As a result the Independent method was expanded upon to develop the Two Step simulation model. With the Two Step, basket sizes are first randomly selected based on the distribution of the training data's transaction lengths. Then, items for each basket are randomly selected with probability weights calculated from the training data.

Evaluation Metric: The KL Divergence

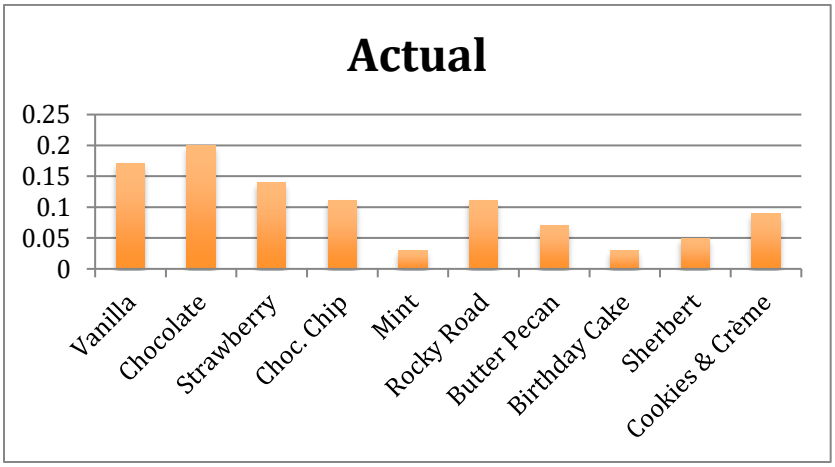
Kullback-Leibler (KL) Divergence measures the difference between two probability distributions of the same variable (Han & Kamber). It measures the information lost when one set of data is used to approximate the other (Han & Kamber). Specifically, suppose that one probability distribution is represented by $p(x)$ and the other is represented by $q(x)$ where x is the variable of interest, the KL Divergence can computed as:

$$D_{KL}(p(x) \parallel q(x)) = \sum_{x \in X} p(x) ln \frac{p(x)}{q(x)}$$

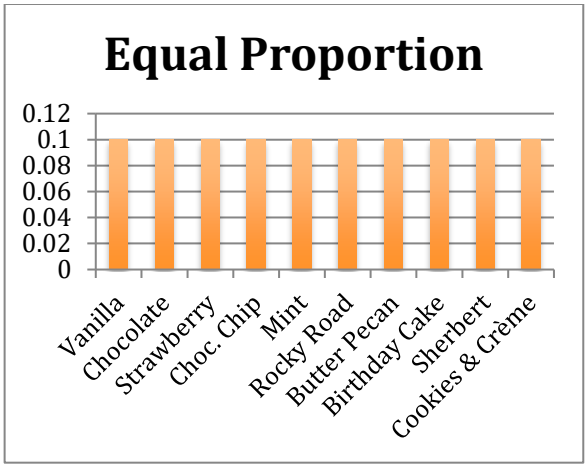
For this research analysis *p(x)* was always the simulated data's probability distribution and *q(x)* was the actual data's probability distribution. Our *x* varied between three different variables for comparison of the simulated data; item frequency grouped by thirty bins of equal width, basket size, and basket value.

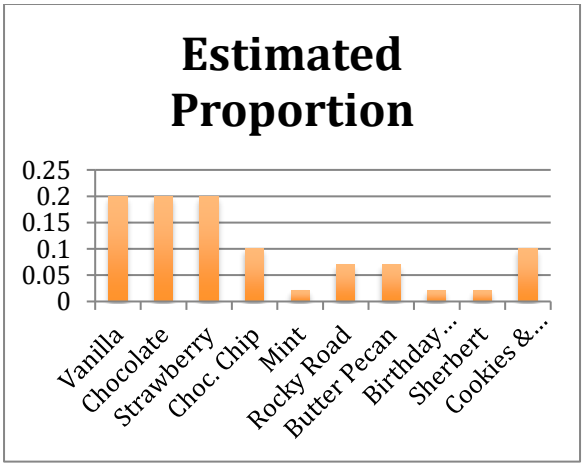
Ideally the KL Divergence is desired to be 0, showing that the simulated and actual data are identical and therefore perfectly approximated from each other. The lower limit of KL divergence is also defined with 0. There is no upper limit for the KL divergence as there is no possible way for two sets of the data to be entirely different from one another in a quantifiable way. The further the KL Divergence value is from 0, the more different the two datasets are from on another.

For example, suppose there is an ice cream shop with 10 flavors and they record the proportion of customers each day that purchase each flavor. The graph of proportions for one randomly selected day is shown below:



The ice cream shop tries to simulate the data using two methods. One assuming each flavor has an equal probability of being chosen. The other following their expected proportions based on the most popular flavors. Both sets of simulated proportion data are shown below:





Using KL Divergence, the Ice Cream Parlor can see which model comes closer to matching the actual proportions seen. Using the equation above, *p(x)* will be the two different simulated proportions and *q(x)* is the actual proportions. The KL Divergence value for the Equal Proportions was 0.1812, while the Estimated Proportions' value was 0.0387. This shows that less information is lost between the Estimated Proportions and the Actual Proportions and therefore it is the better simulation model; as to be expected based on the comparison of the bar plots.

Although the computed value is the distance between the two distributions based on their differences, it is not a distance measure because it is not symmetric (Han & Kamber). If we were to switch the *p(x)* in the equation to be the actual data's probability distribution and the *q(x)* in the equation to be the simulated data's probability distribution the KL Divergence value would be different. This is because "the KL divergence measures the expected number of extra bits required to code samples from *p(x)* when using a code based on *q(x)*" (Han & Kamber). When the frequency distributions are flipped then the difference in code between the two is altered.

The entropy package in R contains a simple function to calculate the KL Divergence called "KL.plugin". It follows the same function as shown above and only requires the input of the two sets of probabilities needing to be compared. The first set is the *p(x)* in the equation and the second listed is the *q(x)*. Optionally one can also include the unit by which the entropy is measured if it needs to be different from the natural units. Example code can be referenced in the appendix of the report.

RESULTS

The Agrawal simulator utilizing item correlation and patterns is too complex for the simulation. It attempts to create relationships between items, when there were no strong item pairs seen in the initial overview of the original data. As a result it does a poor job at accurately simulating the holdout data in regards to every distribution of comparison. Between the two established simulation models, the one that considers all items to be independent, labeled Independent, actually performs better. It does not attempt to create relationships that are not there and as a result follows a basic statistics rule of keeping a model as simple as possible while maintaining accuracy. By modifying the independent method to sample baskets based on the known distribution of the training data's transaction sizes and then fill those baskets with items randomly considering their frequency probability, we created a Two Step method, which outperforms the previous two.

Item Frequency Distribution

Item Frequency describes the support, or number of times the item is seen, within all the transactions of a dataset. The original data contains 92,339 unique items, all available to be found in any transaction.

One of the flaws of the Agrawal and Independent simulators is both re-label items with generic item ID values from 1 to the length of unique items. The items' UPCs from the

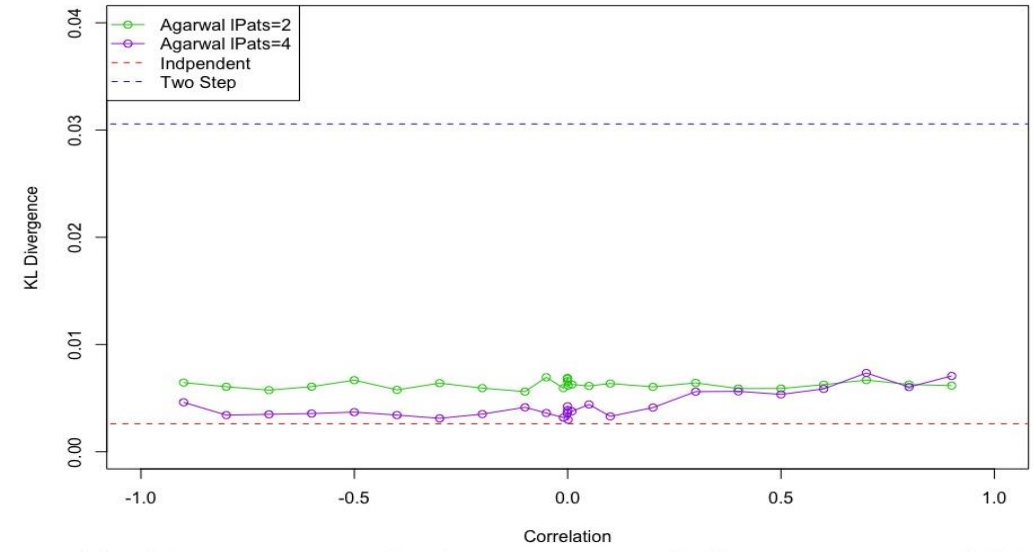
original data's Product_ID labels were used to re-label the simulated data from the Independent and Agrawal methods to match the Product_IDs of the actual holdout data.

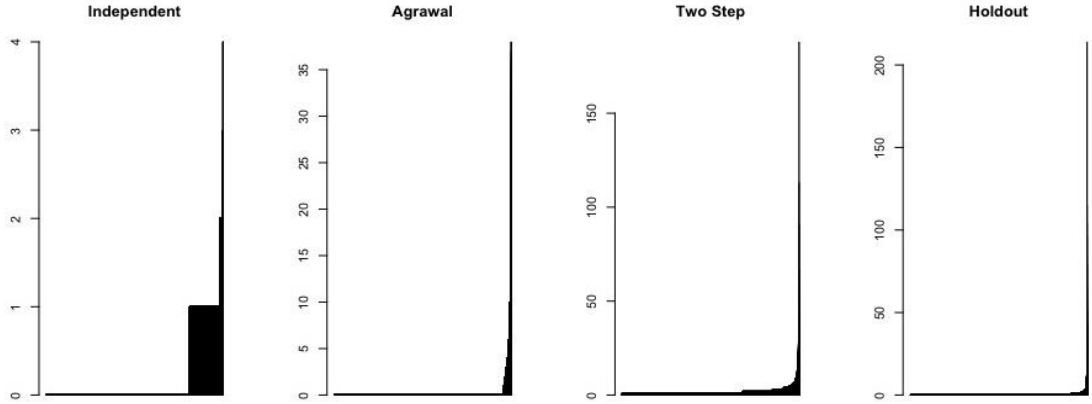
The item frequencies were broken into thirty equal size bins utilizing the minimum and maximum values of the actual holdout data's item frequencies. Here the range was from 0 to 214 across the items, making the bin width 7.37931. It is important that the bin width was consistent across the datasets being compared to ensure that the KL Divergence was an accurate measure of the difference distributions between the actual and simulated transactions.

The KL Divergence between the Independent data and the actual was 0.002602502. The KL Divergence between the Agrawal data and the actual varied between approximately 0.003 and 0.007, with the pattern mean of 4 performing better than the pattern mean of 2 at every correlation except 0.7 and 0.9. The KL Divergence between the Two Step data and the actual was 0.03211682.

This did not match the initial hypothesis that the Agrawal method would perform better than the Independent, as to be expected after the more in-depth look into the original data's lack of correlations between items. The updated though process that the patterns and correlations are simply not strong enough to be represented by any sort of model that includes them is represented by the lack of accuracy seen in the Agrawal method. Even though the Independent assumes consistency across items in terms of their frequency, this means that all of the frequencies are low, matching more closely the distribution of the actual item frequencies.

Picking items based on their probabilities seen in the holdout leads to the best match for the shape of the item frequencies. Unfortunately this is not reflected in the KL divergence because the Two Step method only includes the items that it pulls into baskets for comparison, while the Independent and Agrawal methods consider all items from the original data regardless of if they are in baskets or not. This means with the Two Step all items with a frequency probability of zero are excluded from the simulation and are therefore missed when comparing to the holdout's item frequencies that do include probabilities equal to zero. A different measure could potentially better represent how the Two Step is more similar to the holdout compared to the established simulators, but for consistency the KL Divergence was used for all comparisons. The accuracy of the Two Step model is slightly shown by the fact that despite missing all the items with frequencies of zero, it performs worse by a KL divergence value of less than 0.03.



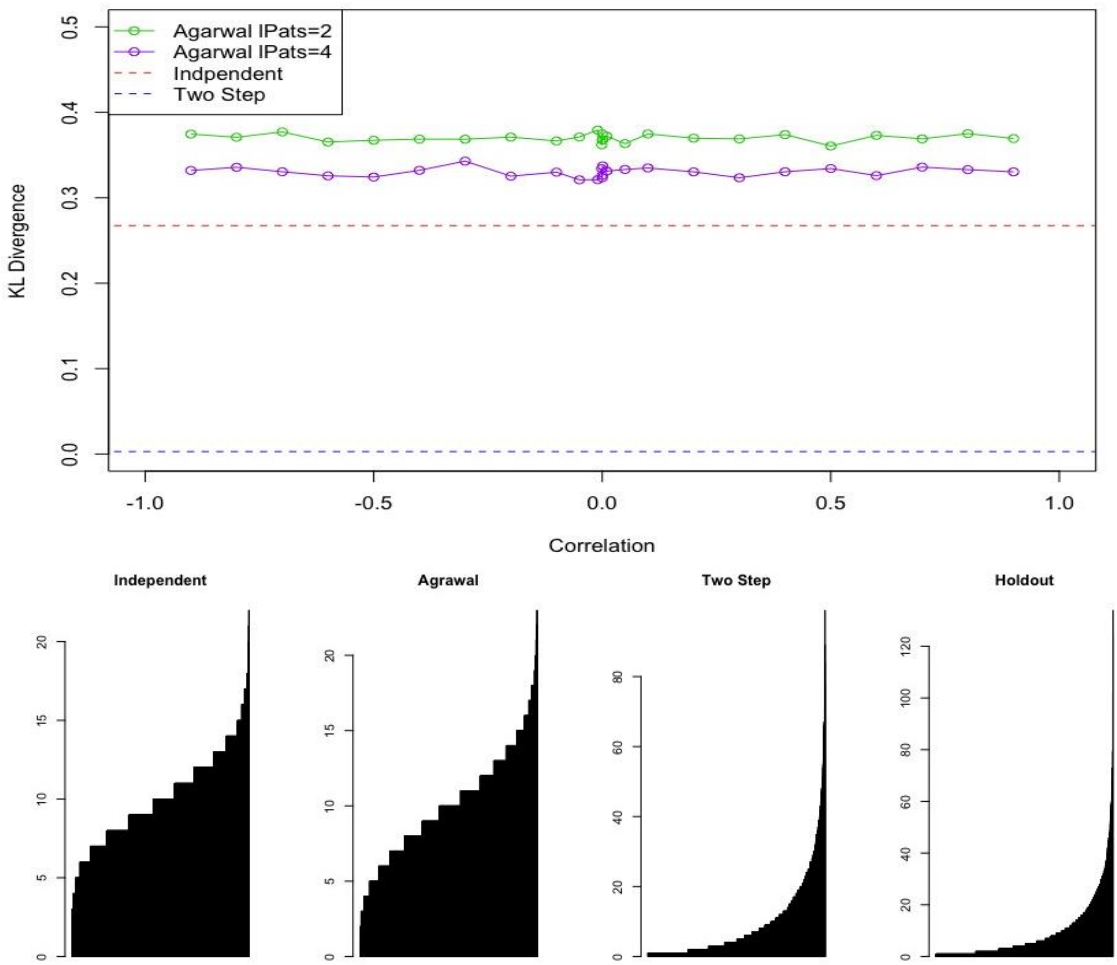


Basket Size Distribution

The basket size distribution describes the number of items seen in each transaction of the dataset. The original data has no upper limit on how many items may be in a transaction and has a lower limit of at least one item per transaction.

The KL Divergence between the independent data and the actual was 0.2725965. The KL Divergence between the Agrawal data and the actual varied between approximately 0.3 and 0.4, with the pattern mean of 4 performing better than the pattern mean of 2. The KL Divergence between the Two Step data and the actual was 0.003628847.

Again, it can be seen the Two Step simulation does significantly better than the Independent and Agrawal methods at accurately simulating the appropriate distribution of the basket sizes. The Holdout follows a distribution similar to an exponential, with the largest baskets containing over 120 items. The Two-Step follows this same distribution and only misses the largest baskets, while the other two methods miss both the distribution as well as all the medium and large basket sizes. The KL Divergence here picks up how much better the Two Step performs with a value extremely close to 0. Not being able to simulate the larger baskets because they assume a Poisson distribution, the Independent and Agrawal have KL Divergence values over 0.25 higher than the Two Step's value.



Basket Value Distribution

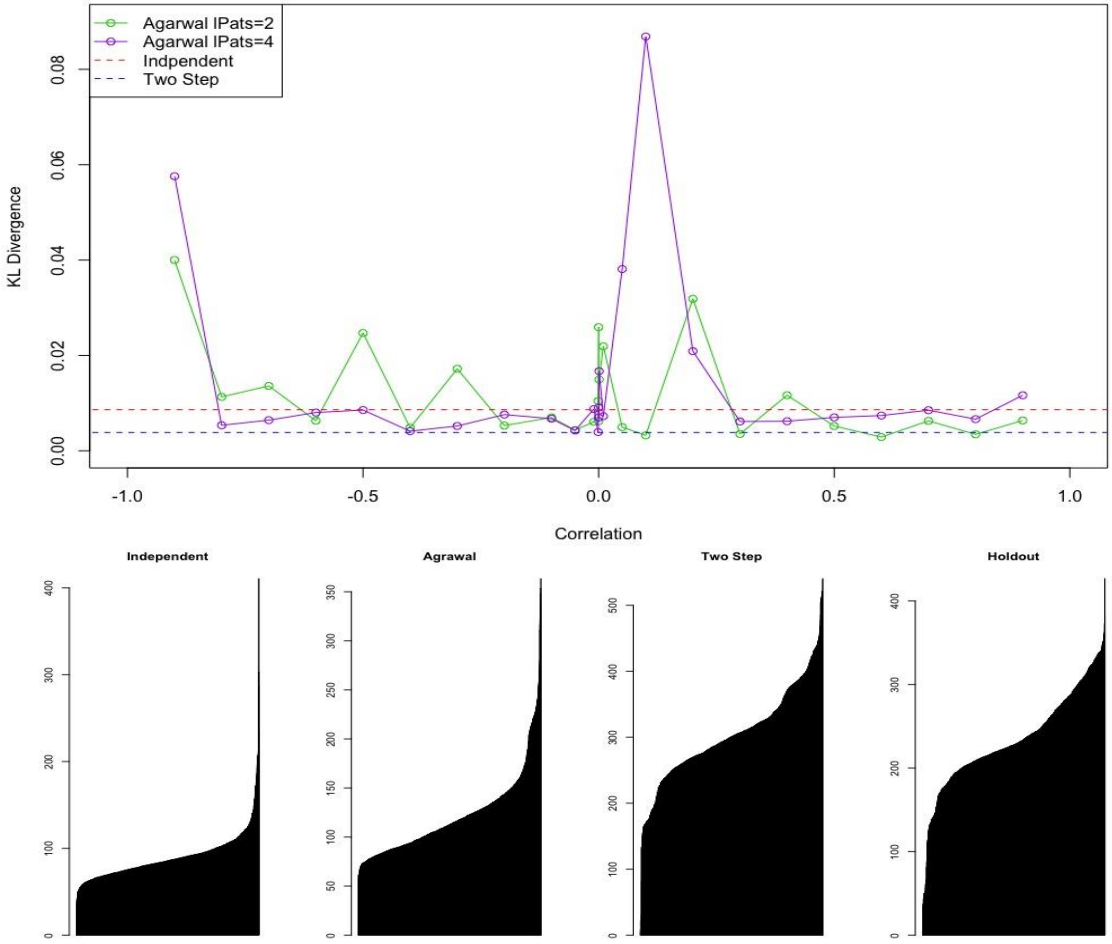
The basket value distribution describes total value of each transaction of the dataset. Values were determined by adding the values of each item in a transaction based on its price calculated in the Prices dataset described earlier. Since the transactions format of the simulations replaces the items' UPCs with generic item labels, they had to be re-labeled back to their UPCs so their prices could be pulled from the Prices dataset. The correctly labeled items from each simulator's output were then saved in a matrix format. A function was created to replace the UPCs in the matrix with their respective prices and then sum all the prices in the basket. These sums of prices represent the values of the baskets that were saved in a vector form for comparison with KL Divergence.

The KL Divergence between the independent data and the actual was 0.02591827. The KL Divergence between the Agrawal data and the actual varied between approximately 0.00 and 0.09, with the pattern mean of 4 and the pattern mean of 2 seeming to perform better and worse than each other depending on the correlation. The KL Divergence between the Two-Step data and the actual was 0.004119917.

The distribution of KL Divergence values here is a lot more varied for the Agrawal method than the previous two areas of comparison and there is no solid explanation as to

why this occurring. One reasoning could be the range is so small that the small changes between correlations are being picked up more noticeably by the graph. The item frequency graph has a range of half the size however and there is hardly any variation with change in correlation. A more realistic explanation is that since the items in each basket determine basket value there is likely to be more variation here since any high or low valued items can really skew the overall basket price. If the Independent and Two Step methods were repeated multiple times we most likely will also see variation among the KL divergence values, just on a smaller scale since they simulate the data more accurately.

On a consistent basis however, the Two-Step still performs better than the Agrawal and the Independent. Plus, with comparisons of the distribution plots one can see that the Two Step best matches the distribution of the holdout out of the three simulators.



CONCLUSIONS

Looking at the distributions of the three points of comparison it is observed that in the three areas of evaluation the Two Step simulation matches the actual data most accurately. It picks up the highs and lows in the same manner of the sorted data and visually looks the most similar in distribution from each set of graphs. This visual conclusion is reflected numerically in the consistently low KL Divergence values for the Two Step regardless of the variable being compared.

Looking at basket size, the KL Divergence values for the Agrawal and the Independent are significantly higher than that of the Two Step method. Although this is the case for the basket value as well, it is on a much smaller scale with a much tighter range of values. We believe this is the case because neither of the established simulations methods consider any aspects of the overall transactions, just the items and whether or not they are correlated. As a result Agrawal and Independent cannot simulate the large baskets because they are restricted to their set distributions as written in the code for. The benefit of the Two Step is the distribution is established by the training data, which in most cases is a broader representation of the holdout data. This is a major correction to a shortcoming of the established methods.

With this improvement, there is also a drawback in the Two Step; it does not represent all possible items while the established methods do. This causes inflation in the KL Divergence between the actual and the Two Step of item frequency. The distribution of the item frequency is best matched by the Two Step's simulation, but without all the items for comparison KL Divergence accounts for the missing information with a higher value. Ideally this would be corrected with an update to the Two Step method that would include all the items regardless of if they are in simulated baskets or not. Considering that this is the only cause of the KL Divergence inflation however, and one can visually see how much better the pattern of item frequencies is matched by the Two Step, then we still feel confident concluding it is the most accurate simulation method.

The basket value comparison is an interesting mix of the observations from the previous two variables of comparison. This may be the case because the calculation of the basket values is not of highest accuracy since the prices of the items were generalized with the idea that they would not vary much over the 102 weeks of data. Since the comparisons are then made on generalizations instead of the truth, it makes sense that there would be more variation here than in the other variables that are exact. The Agrawal most noticeably shows this where the different correlations seem to strongly influence the KL Divergence despite that not being the case in the previous comparisons. Despite the currently unexplainable cause of the Agrawal variation, the Two Step simulation consistently is the most accurate representation of the holdout data. Plus, if one again looks at the distributions, the Two Step continues to the best choice of simulator because it does not try to model low-priced baskets that are not seen in the actual data.

Seeing how the Two Step method performs well across the three variables of comparison affirms that it is the best way to simulate retail data as well as reaffirms the validity of the points made in the "Our Transaction Simulator" section above. With retail data containing a large number of unique items, there are too many possible combinations of items within transactions to have any strong item pair correlations worth considering in simulation. As a result the Agrawal method is too complicated, but the Independent method without any correlation among items is then too simple because it assumes uniformity, which is rarely seem in the real world of retail.

By adding the element of proportions to the Independent method we have found a significant improvement to the established market basket simulators. By considering further points of study described below, this Two Step method can continue to be improved in accuracy as well as the possibilities of use in business.

FURTHER STUDY

In addition to the improvement of the Two-Step method including all items, including ones with frequencies of zero, there are multiple other variables in the data set that could be evaluated to further improve the simulation model. As mentioned in the description of the data there are multiple linkable data files and variables within the main data file, which were ignored for this study. While this allowed for a more broad look at the simulation models, it also can be noted as a limitation to this research.

There is a model for buying decisions that included three steps of analyzing choice of items in a basket, differences between the people buying the items, and then looking at priors for unknown parameters (Manchanda, Ansari, Gupta). This could potentially be adapted into the Two Step model to create a stronger and more accurate simulation with the addition of a third step in the sequence. An interesting point we considered including at the beginning of the research was the effect of promotions on item frequency, basket size and basket value. The retail discount and customer coupon use for the original data is provided, offering two different areas of evaluation for the promotion effect. This could help businesses better analyze how and when to promote items to best improve their weekly profits.

The Two Step model could also be modified to simulate transactions by customer or store location instead of by week. Businesses would then have the ability to expand the range of their simulation opportunities to fit a wider range of research questions. Businesses could also combine simulators to get a more in depth analysis of areas of interest, such as simulating individual stores by week or guessing what a customer's transaction would look like at a new store based on their known transactions in another store.

Ideally there would also be an evaluation of if the KL Divergence is the best method of quantitative comparison. While the KL divergence does show the similarity between data distributions, the lack of interpretable values along with the inaccuracy seen in the item frequency comparisons make it less than ideal. Since the KL Divergence is not a distance there is no quantification of how much worse one simulation does compared to another. This leads to vague conclusions of results and no basis to determine a model's significance of accuracy. An alternate measure that could provide a clearer picture of the difference between simulation methods and the actual data would be preferable. This could also possibly eliminate the need to sort the data for accurate comparison, resulting in a smoother and more straightforward simulation comparison process.

References

Agrawal & Srikant (1994). Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases* (VLDB), pages 487–499, Santiago, Chile.

Agrawal, Imielinski, & Swami (1993). Database mining: a performance perspective. In: *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pages 914-925.

Besanko, Dube, Gupta (2003). Competitive Price Discrimination Strategies in a Vertical Channel Using Aggregate Retail Data. In: *Management Science*, vol. 49, issue 9, pages 1121-1138.

Hahsler, Hornik, & Reutterer(2006). Implications of probabilistic data modeling for mining association rules. In: *From Data and Information Analysis to Knowledge Engineering, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 598–605. Springer-Verlag.

Han & Kamber. (2006). *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers.

Laney, Douglas (2001). *Application Delivery Strategies*. META Group.

Manchanda, Ansari & Gupta (1999) *The "Shopping Basket": A Model for Multicategory Purchase Incidence Decisions*. Marketing Science 18(2):95-114.

Russell & Kamakura (1997). Modeling multiple category brand preference with household basket data. In: *Journal of Retailing*, vol. 73, issue 4, pages 439-461.

Appendix - Copy of Research R Code

```
library(arules)
library(entropy)
library(sqldf)

############################
#### Data Preparation ####
############################

DH <- read.csv("../data/DH/1_original/CSV/transaction_data.csv")
head(DH)

## Create a transactions data object for Arules
DH_list <- split(DH[,"PRODUCT_ID"],  # ItemID
            DH[,"BASKET_ID"])   # TransID
DH_transactions <- as(DH_list,"transactions")

class(DH_transactions)
# A side benefit for creating the transactions data object so early is that all product IDs are registered in the data dictionary.

############################
#### Data Partitioning ####
############################

# Randomly select a week for hold out

#WeekID <- sample(unique(DH$WEEK_NO),1)
HoldoutWeekID <- 12 # hold-out week of our choice

# ----- Identifying the WeekID of each transaction -----

# List of all transaction IDs in the same order of DH_transactions
DH_transIDs <- names(DH_list)

# First, we will find the week-transID correspondence
TransWeek <- sqldf("SELECT DISTINCT WEEK_NO, BASKET_ID FROM DH")

# Then, we subset all transactions that happened in a particular week
HoldoutBaskets <- subset(TransWeek, WEEK_NO==HoldoutWeekID)

nrow(HoldoutBaskets) # number of hold-out baskets
nrow(TransWeek)-nrow(HoldoutBaskets) # number of training baskets

# ----- Splitting transactions into the training and Hold-out Sample Baskets -----
#Holdout Baskets
SelectedTrans <- which(DH_transIDs %in% HoldoutBaskets$BASKET_ID)
DH_trans_holdout   <- DH_transactions[SelectedTrans]

#Training Baskets
UnselectedTrans <- setdiff(1:length(DH_transactions), SelectedTrans)
DH_trans_train <- DH_transactions[UnselectedTrans]

########################################
#### Data Simulation for Holdout Week ####
########################################

# number of unique items possible -- all possible items considered
numitems.all <- nrow(DH_transactions@itemInfo)

# number of transactions to simulate -- equals # trans in the holdout week
bSizes_actual <- size(DH_trans_holdout)
numtrans.ho <- length(bSizes_actual)

# ----- Independent Method -----

independentEXAMPLE <- random.transactions(numitems.all, numtrans.ho,method="independent") #Assuming default values

# ----- Agrawal Method -----
```

```
patterns <- random.patterns(numitems.all) #Assuming default values
agrawalEXAMPLE <- random.transactions(numitems.all, numtrans.ho, method="agrawal", patterns=patterns)

# ----- Two Step Method -----

# Step 1: simulate basket sizes
# Step 2: draw items for each basket

#Find out basket size distribution in training set
bSizes_train <- size(DH_trans_train)

itemFreq_train <- itemFrequency(DH_trans_train, type="absolute") #Frequencies of the items in the holdout
itemFreq_train_prob <- itemFreq_train / sum(itemFreq_train) #Calculate probabilities of the items in the training data

#Sizes of each basket (to be simulated)
bSizes_2step <- sample(bSizes_train, numtrans.ho, replace=T)
simu_df <- NULL

for(i in 1:numtrans.ho){
   pick_items <- bSizes_2step[i]
   #Randomly pick these many items
   item_idx <- sample(1:length(itemFreq_train), pick_items,
                prob=itemFreq_train_prob, replace=T)
   #Create a date frame of the randomly selected items and their corresponding transactions
   twostep <- data.frame(BASKET_ID=i,
                  PRODUCT_ID=item_idx)
   #Combine the dataset
   simu_df <- rbind(simu_df,twostep)
}
#Put the simulation into the transactions format
twostep_list <- split(simu_df["PRODUCT_ID"],  # ItemID
                 simu_df["BASKET_ID"])   # TransID
twostep_trans <- as(twostep_list,"transactions")


######################
#### Evaluations ####
######################

# ----- Item Frequency Comparisons -----

#Create vectors of item frequencies
itemFreq_actual <- itemFrequency(DH_trans_holdout, type="absolute")
itemFreq_simuI <- itemFrequency(independentEXAMPLE, type="absolute")
itemFreq_simuA <- itemFrequency(agrawalEXAMPLE, type="absolute")
itemFreq_simu2 <- itemFrequency(twostep_trans, type="absolute")

#Comparisons of the distributions of sorted item frequencies
par(mfrow=c(1,4))
barplot(sort(itemFreq_simuI), main="Independent")
barplot(sort(itemFreq_simuA), main="Agrawal")
barplot(sort(itemFreq_simu2), main="Two Step")
barplot(sort(itemFreq_actual), main="Actual")
par(mfrow=c(1,1))

#Determine breaks for the KL Divergence computations
all_freqs <- c(itemFreq_actual,itemFreq_simuI,itemFreq_simuA,itemFreq_simu2)
breaks <- seq(min(itemFreq_actual),max(itemFreq_actual),length.out=50)

#Change the item frequecies into probabilities by the breaks
itemFreq_actual_distr <- hist(itemFreq_actual, breaks=breaks, plot = FALSE)$counts
itemFreq_simuI_distr  <- hist(itemFreq_simuI, breaks=breaks, plot = FALSE)$counts
itemFreq_simuA_distr  <- hist(itemFreq_simuA, breaks=breaks, plot = FALSE)$counts
itemFreq_simu2_distr  <- hist(itemFreq_simu2, breaks=breaks, plot = FALSE)$counts

# ----- Basket Size Comparisons -----

#Create vectors of basket sizes
```

```
bSizes_actual <- size(DH_trans_holdout)
bSizes_indep <- size(independentEXAMPLE)
bSizes_agrawal <- size(agrawalEXAMPLE)
bSizes_2step <- size(twostep_trans)

#Comparisons of the distributions of sorted basket sizes
par(mfrow=c(1,4))
barplot(sort(bSizes_indep), main="Independent")
barplot(sort(bSizes_agrawal), main="Agrawal")
barplot(sort(bSizes_2step), main="Two Step")
barplot(sort(bSizes_actual), main="Actual")
par(mfrow=c(1,1))

# ----- Basket Value Comparisons -----

#Create Prices dataset for items
DH$Price <- DH$SALES_VALUE/DH$QUANTITY #Create price for each item individually
DH$Price[is.nan(DH$Price)] <- 0.01 #Change any value of NaN to a penny
DH$Price[is.infinite(DH$Price)] <- 0.01 #Change any value of infinity to a penny
Prices <- aggregate(Price~PRODUCT_ID, data=DH,mean) #Create general prices by the mean of all the individual prices

#Convert Agrawal and Independent items IDs to UPCs
item_labels <- DH_transactions@itemInfo
iLabels <- itemLabels(DH_transactions)

list <- LIST(DH_trans_holdout, decode = FALSE)
  baskets <- list
  list <- decode(list, itemLabels = iLabels)
  baskets <- decode(baskets, itemLabels = iLabels)
  HO_baskets <- as(baskets,"matrix")

list2 <- LIST(agrawalEXAMPLE, decode = FALSE)
  baskets2 <- list2
  list2 <- decode(list2, itemLabels = iLabels)
  baskets2 <- decode(baskets2, itemLabels = iLabels)
  AGR_baskets <- as(baskets2,"matrix")

list3 <- LIST(independentEXAMPLE, decode = FALSE)
  baskets3 <- list3
  list3 <- decode(list3, itemLabels = iLabels)
  baskets3 <- decode(baskets3, itemLabels = iLabels)
  IND_baskets <- as(baskets3,"matrix")

list4 <- LIST(twostep_trans, decode = FALSE)
  baskets4 <- list4
  list4 <- decode(list4, itemLabels = iLabels)
  baskets4 <- decode(baskets4, itemLabels = iLabels)
  TwoStep_baskets <- as(baskets4,"matrix")

#Get Basket Prices
price_ho <- c()
HObaskettotals <- c()
for(i in 1:length(HO_baskets)){
  D <- (HO_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
    itemnum <- which(Prices$PRODUCT_ID == D[d])
    price_ho[d] <- Prices$Price[itemnum]
  }
  HObaskettotals[i] <- sum(price_ho)
}

price_agr <- c()
agrbaskettotals <- c()
for(i in 1:length(AGR_baskets)){
  D <- (AGR_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
    itemnum <- which(Prices$PRODUCT_ID == D[d])
```

```
   price_agr[d] <- Prices$Price[itemnum]
  }
  agrbaskettotals[i] <- sum(price_agr)
}

price_ind <- c()
indbaskettotals <- c()
for(i in 1:length(IND_baskets)){
  D <- (IND_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
    itemnum <- which(Prices$PRODUCT_ID == D[d])
    price_ind[d] <- Prices$Price[itemnum]
  }
  indbaskettotals[i] <- sum(price_ind)
}

price_twostep <- c()
twostepbaskettotals <- c()
for(i in 1:length(TwoStep_baskets)){
  D <- (TwoStep_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
    itemnum <- which(Prices$PRODUCT_ID == D[d])
    price_twostep[d] <- Prices$Price[itemnum]
  }
  twostepbaskettotals[i] <- sum(price_twostep)
}

#Comparisons of the distributions of sorted basket prices
par(mfrow=c(1,4))
barplot(sort(indbaskettotals), main="Independent")
barplot(sort(agrbaskettotals), main="Agrawal")
barplot(sort(twostepbaskettotals), main="Two Step")
barplot(sort(HObaskettotals), main="Holdout")
par(mfrow=c(1,1))

########################
#### Batch Process ####
########################

#Number of unique items possible in the holdout
numitems.ho <- nrow(DH_trans_holdout@itemInfo)

#Number of transactions to simulate (equals # trans in the holdout week)
bSizes.ho_actual <- size(DH_trans_holdout)
numtrans.ho <- length(bSizes.ho_actual)

#Number of transactions in the training
bSizes_train <- size(DH_trans_train)

#Frequencies of the items in the holdout
itemFreq_train <- itemFrequency(DH_trans_train, type="absolute")
itemFreq_train_prob <- itemFreq_train / sum(itemFreq_train)

####################
#### Item Freq ####
####################

#Independent Item Frequency
item_frequency_ind <- function(num_items,num_trans,actual_item_freqs){
   ind <- random.transactions(num_items,num_trans,method="independent")
   numitemsI <- itemFrequency(ind, type="absolute")
   breaks <- seq(min(actual_item_freqs),max(actual_item_freqs),length.out=30)
   itemFreq_actual_distr <- hist(actual_item_freqs, breaks=breaks, plot = FALSE)$counts
   itemFreq_simuI_distr  <- hist(numitemsI, breaks=breaks, plot = FALSE)$counts
   KL_item_ind <- KL.plugin(sort(itemFreq_simuI_distr),sort(itemFreq_actual_distr))
   KL_item_ind
}
```

```
#Agrawal Item Frequency, lPats = 2
item_frequency_agrawal2 <- function(num_items,num_trans,actual_item_freqs, corr){
   patterns <- random.patterns(num_items,corr=corr, lPats=2)
   agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
   numitemsA <- itemFrequency(agr, type="absolute")
   breaks <- seq(min(actual_item_freqs),max(actual_item_freqs),length.out=30)
   itemFreq_actual_distr <- hist(actual_item_freqs, breaks=breaks, plot = FALSE)$counts
   itemFreq_simuA_distr  <- hist(numitemsA, breaks=breaks, plot = FALSE)$counts
   KL_item_agr <- KL.plugin(sort(itemFreq_simuA_distr),sort(itemFreq_actual_distr))
   KL_item_agr
}

#Agrawal Item Frequency, lPats = 4
item_frequency_agrawal4 <- function(num_items,num_trans,actual_item_freqs, corr){
  patterns <- random.patterns(num_items,corr=corr, lPats=4)
  agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
  numitemsA <- itemFrequency(agr, type="absolute")
  breaks <- seq(min(actual_item_freqs),max(actual_item_freqs),length.out=30)
  itemFreq_actual_distr <- hist(actual_item_freqs, breaks=breaks, plot = FALSE)$counts
  itemFreq_simuA_distr  <- hist(numitemsA, breaks=breaks, plot = FALSE)$counts
  KL_item_agr <- KL.plugin(sort(itemFreq_simuA_distr),sort(itemFreq_actual_distr))
  KL_item_agr
}

#Two Step Item Frequency
item_frequency_twostep <- function(training_basket_sizes,ho_trans,actual_item_freqs){
bSizes_2step <- sample(training_basket_sizes, ho_trans, replace=T)
simu_df <- NULL
for(i in 1:ho_trans){
  pick_items <- bSizes_2step[i]
  item_idx <- sample(1:length(itemFreq_train), pick_items,
                  prob=itemFreq_train_prob, replace=F)
  twostep <- data.frame(BASKET_ID=i,
                    PRODUCT_ID=item_idx)
  simu_df <- rbind(simu_df,twostep)
}
twostep_list <- split(simu_df[,"PRODUCT_ID"],  # ItemID
                  simu_df[,"BASKET_ID"])   # TransID
twostep_trans <- as(twostep_list,"transactions")
numitems2 <- itemFrequency(twostep_trans, type="absolute")
breaks <- seq(min(actual_item_freqs),max(actual_item_freqs),length.out=30)
itemFreq_actual_distr <- hist(actual_item_freqs, breaks=breaks, plot = FALSE)$counts
itemFreq_simu2_distr  <- hist(numitems2, breaks=breaks, plot = FALSE)$counts
KL_item_2step <- KL.plugin(sort(itemFreq_simu2_distr),sort(itemFreq_actual_distr))
KL_item_2step
}

itemFreq_actual <- itemFrequency(DH_trans_holdout, type="absolute")

#Independent Item Frequency KL Divergence
KL0 <- item_frequency_ind(numitems.ho,numtrans.ho,itemFreq_actual)

#Calculate KL Divergence for different correlations of Agrawal
x <- c(seq(-0.9,-0.1,0.1),-0.05,-0.01,-0.001,-0.0001,0.0001,0.001,0.01,0.05,seq(0.1,0.9,0.1))
y <- c()
for(i in 1:length(x)){
   cor_val <- x[i]
   y[i] <- item_frequency_agrawal2(numitems.ho,numtrans.ho,itemFreq_actual,cor_val)
}

yy <- c()
for(i in 1:length(x)){
  cor_val <- x[i]
  yy[i] <- item_frequency_agrawal4(numitems.ho,numtrans.ho,itemFreq_actual,cor_val)
}

#Two Step Item Frequency KL Divergence
KL2step <- item_frequency_twostep(bSizes_train,numtrans.ho,itemFreq_actual)
```

```
#Plot the Item Frequency KL Divergence for comparison
plot(x,y,xlab="Correlation",ylab="KL Divergence",ylim=c(0.0,0.04),xlim = c(-1,1),type="o",col=3)
lines(x,yy,type = "o", col="purple")
lines(c(-1.1,1.1),c(KL0,KL0),col="red",lty=2)
lines(c(-1.1,1.1),c(KL2step,KL2step),col=4,lty=2)
legend("topleft",c("Agarwal lPats=2","Agarwal lPats=4","Indpendent","Two Step"),
       lty=c(1,1,2,2),
       pch=c(1,1,NA,NA),
       col=c(3,"purple",2,4))

######################
#### Basket Size ####
######################

#Independent Basket Size
basket_size_ind <- function(num_items,num_trans,Sizes_actual){
        ind <- random.transactions(num_items,num_trans,method="independent")
        basket <- size(ind)
        KL_basket_ind <- KL.plugin(sort(basket),sort(Sizes_actual))
        KL_basket_ind
 }

#Agrawal Basket Size, lPats = 2
basket_size_agr2 <- function(num_items,num_trans,Sizes_actual,corr){
  patterns <- random.patterns(num_items,corr=corr,lPats=2)
  agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
  basket <- size(agr)
  KL_basket_agr <- KL.plugin(sort(basket),sort(Sizes_actual))
  KL_basket_agr
}

#Agrawal Basket Size, lPats = 4
basket_size_agr4 <- function(num_items,num_trans,Sizes_actual,corr){
  patterns <- random.patterns(num_items,corr=corr,lPats=4)
  agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
  basket <- size(agr)
  KL_basket_agr <- KL.plugin(sort(basket),sort(Sizes_actual))
  KL_basket_agr
}

#Two Step Basket Size
basket_size_twostep <- function(training_basket_sizes,ho_trans,Sizes_actual){
  bSizes_2step <- sample(training_basket_sizes, ho_trans, replace=T)
  simu_df <- NULL
  for(i in 1:ho_trans){
    pick_items <- bSizes_2step[i]
    item_idx <- sample(1:length(itemFreq_train), pick_items,
                       prob=itemFreq_train_prob, replace=T)
    twostep <- data.frame(BASKET_ID=i,
                          PRODUCT_ID=item_idx)
    simu_df <- rbind(simu_df,twostep)
  }
  twostep_list <- split(simu_df["PRODUCT_ID"],  # ItemID
                        simu_df["BASKET_ID"])   # TransID
  twostep_trans <- as(twostep_list,"transactions")
  basket <- size(twostep_trans)
  KL_basket_2step <- KL.plugin(sort(basket),sort(Sizes_actual))
  KL_basket_2step
}

#Independent Basket Size KL Divergence
KLB0 <- basket_size_ind(numitems.ho,numtrans.ho,bSizes_actual)

#Calculate KL Divergence for different correlations of Agrawal
x <- c(seq(-0.9,-0.1,0.1),-0.05,-0.01,-0.001,-0.0001,0.0001,0.001,0.01,0.05,seq(0.1,0.9,0.1))
y2 <- c()
for(i in 1:length(x)){
  cor_val <- x[i]
```

```
  y2[i] <- basket_size_agr2(numitems.ho,numtrans.ho,bSizes_actual,cor_val)
} #LOWEST = 0.05, 0.7175684

y4 <- c()
for(i in 1:length(x)){
  cor_val <- x[i]
  y4[i] <- basket_size_agr4(numitems.ho,numtrans.ho,bSizes_actual,cor_val)
}

#Two Step Basket Size KL Divergence
KLB2step <- basket_size_twostep(bSizes_train,numtrans.ho,bSizes_actual)

#Plot the Basket Size KL Divergence for comparison
plot(x,y2,xlab="Correlation",ylab="KL Divergence",ylim=c(0.0,0.50),xlim = c(-1,1),type="o", col=3)
lines(x,y4,type="o",col="purple")
lines(c(-1.1,1.1),c(KLB0,KLB0),col="red",lty=2)
lines(c(-1.1,1.1),c(KLB2step,KLB2step),col=4,lty=2)
legend("topleft",c("Agarwal lPats=2","Agarwal lPats=4","Indpendent","Two Step"),
       lty=c(1,1,2,2),
       pch=c(1,1,NA,NA),
       col=c(3,"purple",2,4))

########################
#### Basket Value ####
########################

#Convert Agrawal and Independent items IDs to UPCs
item_labels <- DH_transactions@itemInfo

#Actual Basket Value for Holdout
price_ho <- c()
HObaskettotals <- c()
for(i in 1:length(HO_baskets)){
  D <- (HO_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
    itemnum <- which(Prices$PRODUCT_ID == D[d])
    price_ho[d] <- Prices$Price[itemnum]
  }
  HObaskettotals[i] <- sum(price_ho)
}

#Independent Basket Value
price <- c()
baskettotals_ind <- c()
basket_value_ind <- function(num_items,num_trans,iLabels,baskettotals_actual){
  ind <- random.transactions(num_items,num_trans,method="independent")
  list <- LIST(ind, decode = FALSE)
  baskets <- list
  list <- decode(list, itemLabels = iLabels)
  baskets <- decode(baskets, itemLabels = iLabels)
  IND_baskets <- as(baskets,"matrix")
  for(i in 1:length(IND_baskets)){
    D<- (IND_baskets[i,])
    D <- unlist(D)
    for(d in 1:length(D)){
      itemnum <- which(Prices$PRODUCT_ID == D[d])
      price[d] <- Prices$Price[itemnum]
    }
    baskettotals_ind[i] <- sum(price)
  }
  KL_basket_ind <- KL.plugin(sort(baskettotals_ind),sort(baskettotals_actual))
  KL_basket_ind
}

#Agrawal Basket Value, lpats=2
price <- c()
baskettotals <- c()
basket_value_agr2 <- function(num_items,num_trans,corr,iLabels,baskettotals_actual){
```

```
 patterns <- random.patterns(num_items,corr=corr,lPats=2)
 agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
 list <- LIST(agr, decode = FALSE)
 baskets <- list
 list <- decode(list, itemLabels = iLabels)
 baskets <- decode(baskets, itemLabels = iLabels)
 AGR_baskets <- as(baskets,"matrix")
 for(i in 1:length(AGR_baskets)){
  D<- (AGR_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
   itemnum <- which(Prices$PRODUCT_ID == D[d])
   price[d] <- Prices$Price[itemnum]
  }
  baskettotals[i] <- sum(price)
 }
 KL_value_agr <- KL.plugin(sort(baskettotals),sort(baskettotals_actual))
 KL_value_agr
}

#Agrawal Basket Value, lpats=4
price <- c()
baskettotals_agr4 <- c()
basket_value_agr4 <- function(num_items,num_trans,corr,iLabels,baskettotals_actual){
 patterns <- random.patterns(num_items,corr=corr,lPats=4)
 agr <- random.transactions(num_items,num_trans,method="agrawal",patterns=patterns)
 list <- LIST(agr, decode = FALSE)
 baskets <- list
 list <- decode(list, itemLabels = iLabels)
 baskets <- decode(baskets, itemLabels = iLabels)
 AGR_baskets <- as(baskets,"matrix")
 for(i in 1:length(AGR_baskets)){
  D<- (AGR_baskets[i,])
  D <- unlist(D)
  for(d in 1:length(D)){
   itemnum <- which(Prices$PRODUCT_ID == D[d])
   price[d] <- Prices$Price[itemnum]
  }
  baskettotals_agr[i] <- sum(price)
 }
 KL_value_agr <- KL.plugin(sort(baskettotals_agr),sort(baskettotals_actual))
 KL_value_agr
}

#Two Step Basket Value
price <- c()
baskettotals_twostep <- c()
basket_value_twostep <- function(training_basket_sizes,ho_trans,iLabels,baskettotals_actual){
 bSizes_2step <- sample(training_basket_sizes, ho_trans, replace=T)
 simu_df <- NULL
 for(i in 1:ho_trans){
  pick_items <- bSizes_2step[i]
  item_idx <- sample(1:length(itemFreq_train), pick_items,
                     prob=itemFreq_train_prob, replace=T)
  twostep <- data.frame(BASKET_ID=i,
                        PRODUCT_ID=item_idx)
  simu_df <- rbind(simu_df,twostep)
 }
 twostep_list <- split(simu_df["PRODUCT_ID"],  # ItemID
                       simu_df["BASKET_ID"])   # TransID
 twostep_trans <- as(twostep_list,"transactions")
 list <- LIST(twostep_trans, decode = FALSE)
 baskets <- list
 list <- decode(list, itemLabels = iLabels)
 baskets <- decode(baskets, itemLabels = iLabels)
 TwoStep_baskets <- as(baskets,"matrix")
 for(i in 1:length(TwoStep_baskets)){
  D<- (TwoStep_baskets[i,])
  D <- unlist(D)
```

```
   for(d in 1:length(D)){
     itemnum <- which(Prices$PRODUCT_ID == D[d])
     price[d] <- Prices$Price[itemnum]
   }
   baskettotals_twostep[i] <- sum(price)
 }
 KL_basket_2step <- KL.plugin(sort(baskettotals_twostep),sort(baskettotals_actual))
 KL_basket_2step
}

#Independent Basket Value KL Divergence
KLV0 <- basket_value_ind(numitems.ho,numtrans.ho,iLabels,HObaskettotals)

#Calculate KL Divergence for different correlations of Agrawal
x <- c(seq(-0.9,-0.1,0.1),-0.05,-0.01,-0.001,-0.0001,0.0001,0.001,0.01,0.05,seq(0.1,0.9,0.1))
yv2 <- c()
for(i in 1:length(x)){
 cor_val <- x[i]
 yv2[i] <- basket_value_agr2(numitems.ho,numtrans.ho,cor_val,iLabels,HObaskettotals)
}

yv4 <- c()
for(i in 1:length(x)){
 cor_val <- x[i]
 yv4[i] <- basket_value_agr4(numitems.ho,numtrans.ho,cor_val,iLabels,HObaskettotals)
}

#Two Step Basket Value KL Divergence
KLV2step <- basket_value_twostep(bSizes_train,numtrans.ho,iLabels,HObaskettotals)

#Plot the Basket Value KL Divergence for comparison
plot(x,yv2,xlab="Correlation",ylab="KL Divergence",ylim=c(0.0,0.09),xlim = c(-1,1),type="o", col=3)
lines(x,yv4,type="o",col="purple")
lines(c(-1.1,1.1),c(KLV0,KLV0),col="red",lty=2)
lines(c(-1.1,1.1),c(KLV2step,KLV2step),col=4,lty=2)
legend("topleft",c("Agarwal lPats=2","Agarwal lPats=4","Indpendent","Two Step"),
       lty=c(1,1,2,2),
       pch=c(1,1,NA,NA),
       col=c(3,"purple",2,4))
```