Mining Big Data – Assignment 3b Report

Students Names and Numbers:

Ebubekir Pulat – a1833363

Chunyu Zhang – a1751743

Xingyuan Zhang – a1808020

Executive Summary

Following the store manager's interesting discovery that certain groupings of products were common in customer baskets, the *West Wing Supermarket* consumer research team has decided to investigate these popular item agglomerations.

More specifically, the team intends to identify frequent item-sets, popular groupings of products in customer baskets. This is to provide evidential guidance for the store's re-design, to increase the proximity of items, belonging to the same frequent item-set, potentially reducing the necessary time to pick up all the products and head to the checkout, increasing the store's sales per minute ratio.

Furthermore, the research team has deduced that it would also be beneficial to experiment with various recommendations methods including collaborative filtering, to generate product recommendations for customers browsing the stores' online catalogue. These recommendations may be presented in the form of a 'recommended products' feature on the website, or the re-ordering of showcased products to place suggested products higher up on a page.

The team intends to source the necessary data and information to determine frequent item-sets and produce recommendations from, through the *West Wing Supermarket Customer Points* scheme, that contains records of customer purchases including facts about the customer, their purchased products, the time, and location.

Although the company is currently operating at approximately 50,000 purchases, online and in-person each day, this project intends to design the system to handle at least 1 million of such purchases daily. This is in consideration of the company's fast growing customer base, market share, supply operations and number of stores around the nation.

The expected outcome of an effective frequent item-set identifier and recommendation system is an increase in sales per day, improvement in customer satisfaction survey scores and a streamlined path from entering the store, to leaving the checkout, to reduce in-store human traffic.

Introduction

The grocery store is looking to increase sales by better understanding customer buying habits with the grocery purchase records dataset. The purpose of this project is to mine and analyse the historical customer purchase data to identify common product combinations (frequent item-sets) and association rules between different products. With this information, the product placement layout will be improved so that the store can reach a higher sales rate.

Furthermore, two recommendation systems, one based on collaborative filtering and another based on an original idea generated by the team, have been experimented with to pick out certain items for users as suggestions. The recommendation systems were then evaluated on their performance. Comments written in this report about the project, may provide direction for future investigations into this issue.

Exploratory Analysis

The top 20 best-selling items have been extracted by applying some counting and sorting methods on the data.

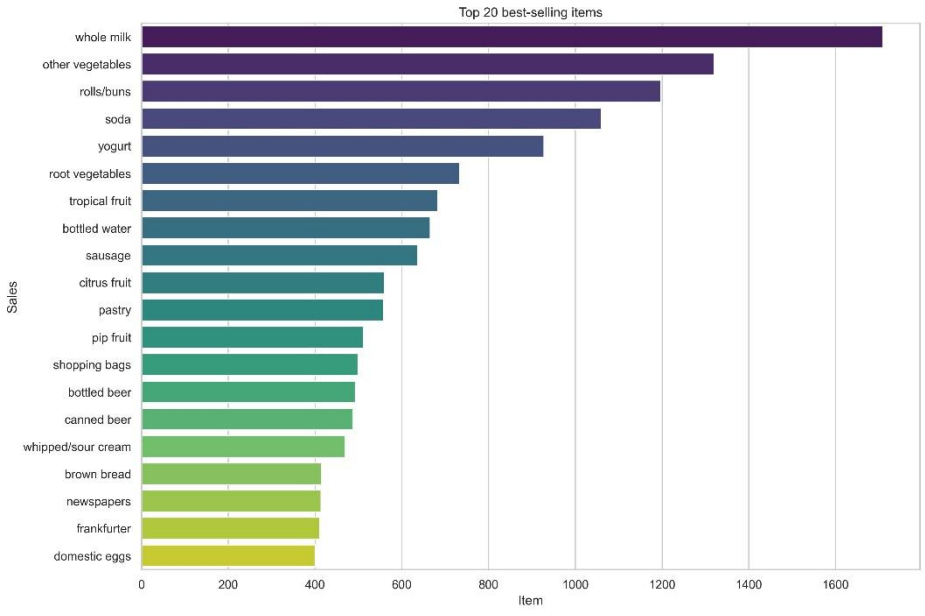

Figure 1. Top 20 best-selling items

In the top 20 best-selling products, we can find that 90% of the best-selling products in the store are food, with whole milk ranking first. Among these foods, fresh foods such as

vegetables, fruits and dairy products account for a high proportion. After digging a little deeper, we counted the customers who visited the store most frequently.

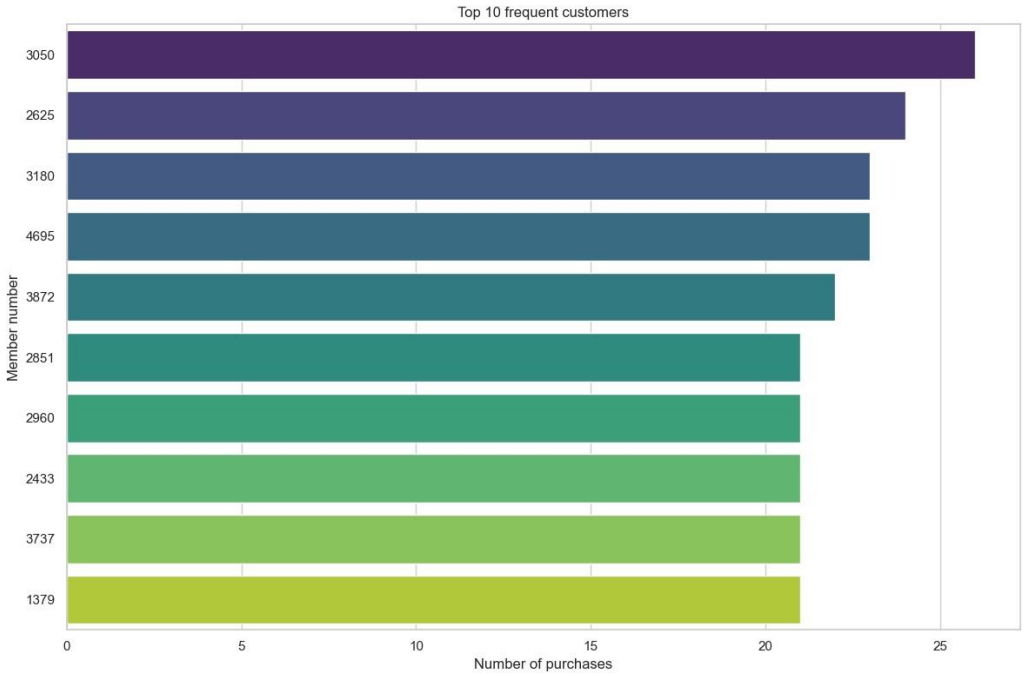

Figure 2. Top 10 frequent customers

The figure above shows the customers who most often make purchases at this store. The grocery store may introduce new promotional strategies to reward customers who always make repeat purchases to encourage more customers to visit.

Frequent Pattern Mining

In this project, we have chosen to use the FP-growth algorithm for frequent pattern mining. This is because our dataset is a relatively large dataset and must be considered even scaled up to one million transactions for computation. And when dealing with large-scale datasets, FP-growth algorithm is a more efficient frequent itemset mining method because it does not require candidate set generation, so FP-growth is more efficient than the traditional Apriori algorithm. In the code, we use the fpgrowth function from Python's mlxtend library to implement this algorithm.

Regarding hyper-parameters, we use 0.05 as minimum support and 0.8 as minimum threshold for confidence at first, but the results are empty. We realized that this is a large retail database containing multiple items and using a lower support is a better

choice. So finally, we used 0.001 as minimum support and 0.05 as minimum threshold so that we got a result with a suitable number of items.

We found that adjusting support and confidence has a very significant effect on mining results by experimenting with different minimum support and confidence. The combined results show that more patterns can be mined when the threshold is low, but these patterns may be with uncertainty. And when the threshold is adjusted higher, the number of patterns in the results decreases, but the generalization of each pattern is stronger.

We first experimented with support. At a support of 0.05 (5%) we could not get any patterns, at a support of 0.005 (0.5%) we got 7 results, at 0.003 (0.3%) we got 31 results and at 0.001 (0.1%) we got 239 results. Lower support can be used to explore less common but potentially beneficial combinations of goods and maximize the results of recommendations.

For confidence we tried 0.5 (50%), 0.1 (10%), and 0.05 (5%), for which we conducted several experiments, including the use of training data and test data. We believe that the best performer is 0.05 because in large supermarket orders with multiple items, too high a level of confidence can lead to ignoring many of the potential items. Also, this value favours a better result when using interest as a parameter for recommendation.

Moreover, we believe that using interest as a parameter is the best way to determine the recommendation results because it combines the performance of minimum support and confidence. When the confidence is 0.05 (5%), we can get a lot of patterns where interest is positive, there are 23.

Due to the nature of supermarkets, the cost of wrong recommendation is very low and once the recommendation is successful, the company will get better profit. At the same time, supermarkets have thousands of members and some of them have niche shopping habits. So, to maximize the profit, we need more referrals and focus on more niche shopping habits.

Collaborative Filtering

The collaborative filtering performed is based on user-similarity and was implemented using a 3-phase approach, normalise the data, find nearest neighbours and recommend the weighted average of those neighbour's choices. To normalise the data, for every row in the user-item matrix, the mean of the row was calculated and every entry in that row was reduced by that mean.

Following, for each user (row), and for each item the user had not bought, the 1000 most similar neighbours who have purchased that product were identified. This was done by calculating which other users resulted in the highest dot product-based similarity scores with the reference user.

Once the neighbours have been determined, the weighted average of their purchase counts was obtained, where the counts of neighbours corresponding to higher similarities, are weighed more. This weighted average was calculated for each non-bought item in the reference user's row, and the item connected to the largest weighted average, was recommended to the reference user. This is as it is hypothesised a higher weighted average corresponds to a higher likelihood of a correct recommendation. Recommendations are provided by updating the associated user item matrix entry, with the calculated weighted average. After finalising the above process, the user item matrix will likely have a new entry, a recommendation, for each user.

Although the method outline sounds logical, it is critical to evaluate its performance. This methodology was evaluated using root mean squared error (RMSE), as the recommendation for each user consists of a weighted average, theoretically ranging from 0 to ∞ (infinity). Based on the methodology's logical reasoning, the higher the weighted average value, the increased likelihood the user would approve of purchasing the given product. This is as if the system predicted a user will buy 2 loafs of bread, and 1 watermelon, clearly the recommendation method believes the user is more likely to buy loafs of bread than watermelons.

RMSE evaluation was applied, by calculated the RMSE across all predicted weighted average values for certain user-item entries, in respect to the actual value, derived from the test set.

Turning Frequent Item-Sets into Recommendations

A simple yet interesting system was developed that takes as input, frequent item sets generated from the frequent pattern mining conducted, and their associated interest values. For each user, the system collects all frequent item-sets that have at least one item the user bought according to the training data. Then for each of these item sets, the corresponding 'item set relevance' is calculated, which is a hypothesised indicator of an item-set's relation to a user.

The 'itemset relevance' is calculated by finding the number of instances the user bought each of the items in the item-set and multiplying each of these counts with the interest of the itemset. Once all itemset relevancies have been calculated, the item-sets are ranked based on this measure, and the itemset ranked first is chosen as the set of recommendations for the given user. Every item in the selected item set, that the user has not bought in the training data, is recommended, and the collection of these sets of recommendations, for each user (where applicable) is the output of the system.

Results

1) For frequent mining, our code uses the "get_conf_supp" function to get the result containing the pattern, support and confidence and prints it out using the "print_frequent" function. We believe that support and confidence should be used to evaluate patterns, but maintaining a high confidence is difficult in large and diverse datasets. So, we should additionally consider engaging the support. High support for items ensures that recommended items are likely to be purchased by most customers.

Regarding the collaborative filtering, RMSE was used for evaluation, by comparing the difference between the recommendations (weighted averages of neighbours' purchases) made by the system, and the number of times the given user bought those recommended items, in the test set. In the collaborative filtering, the recommendations (item scores) were ordered and selected based on the magnitude of their corresponding weighted average. The weighted average for an item is the mean, of the sum, of the products between a neighbour's similarity to the user, and the number of times they purchased the specified item in the train set; this calculation is performed for the 1000 neighbours most alike to the being-recommended-to user.

In terms of the recommendations deduced from the generated frequent item-sets, these were assessed using the accuracy measure. This accuracy simply represents the number of recommended items that were bought by the given user in the test data, over the count of all recommendations, where a recommendation refers to a single item. These recommendations were ordered based on their calculated item-set relevancies (as described under 'Turning Frequent Item-Sets into Recommendations').

2) It has been sorted using support as a metric, and the five patterns with the highest support values are printed here:

```
5 patterns of train dataset:
patterns:[whole milk, rolls/buns], support: 0.007913099776994462, confidence: 0.06703229737964655
patterns:[other vegetables, whole milk], support: 0.006977915257895115, confidence: 0.07519379844961241
patterns:[yogurt, whole milk], support: 0.006546291633695417, confidence: 0.10178970917225949
patterns:[soda, whole milk], support: 0.00597079346809582, confidence: 0.08058252427184466
patterns:[soda, other vegetables], support: 0.005467232573196173, confidence: 0.07378640776699029
```

Figure 3. Five examples of frequent patterns on training set

```
5 patterns of test dataset:
patterns:[other vegetables, whole milk], support: 0.0033055967172005017, confidence: 0.0506108202443281
patterns:[tropical fruit, whole milk], support: 0.0020517496865382423, confidence: 0.0518731988472625
patterns:[bottled water, other vegetables], support: 0.0015958053117519663, confidence: 0.052631578947368425
patterns:[pastry, whole milk], support: 0.0013678331243588283, confidence: 0.05309734513274337
patterns:[beef, whole milk], support: 0.0011398609369656903, confidence: 0.06896551724137932
```

Figure 4. Five examples of frequent patterns on test set

3) 10 Examples of itemset-derived recommendations

```
Recommendations for User 1032
grapes

Recommendations for User 1033
processed cheese
rolls/buns

Recommendations for User 1034
UHT-milk
tropical fruit

Recommendations for User 1035
processed cheese
rolls/buns

Recommendations for User 1037
processed cheese
rolls/buns
```

```
Recommendations for User 1038
napkins
pastry

Recommendations for User 1039
processed cheese
rolls/buns

Recommendations for User 1040
processed cheese
rolls/buns

Recommendations for User 1041
frozen vegetables
sausage

Recommendations for User 1042
butter
bottled water
```

4) Figures 5 and 6 are two bar charts of metrics obtained, one set indicating the support of the frequent patterns and the other set indicating the confidence of the corresponding patterns.

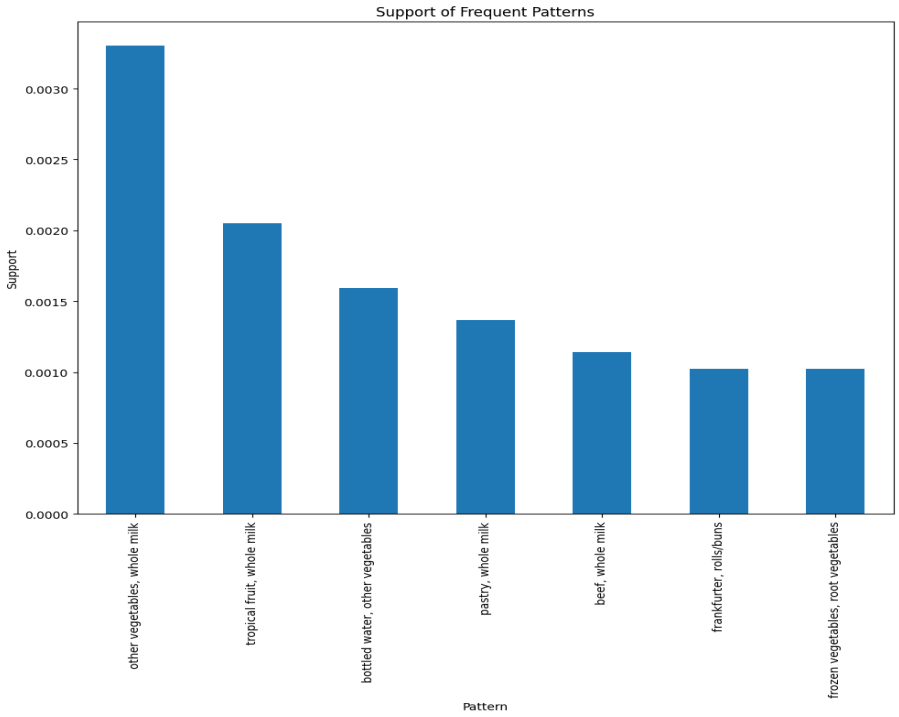

Figure 5. Bar charts of support and frequent patterns

Patterns with higher support appear more frequently in the dataset, such as "other vegetables" and "whole milk". However, the confidence for this pattern is lower, suggesting that while these two items are not strongly correlated, it is simply that this combination of items is a more common purchase option. For patterns with high support, our company can consider bundling sales, promotional activities, or optimizing the placement of these items to further improve sales efficiency and customer satisfaction.

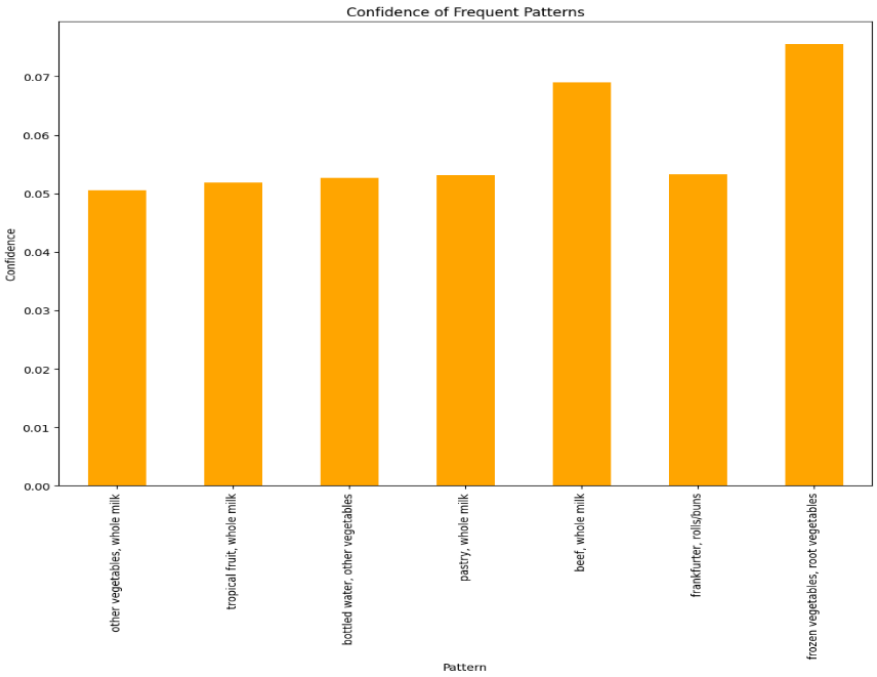

Figure 6. Bar charts of confidence and frequent patterns

The confidence responds to the probability of purchasing the latter in the antecedent purchase condition. For patterns with higher confidence, for example, “beef” and “whole milk” does not have a high support, this may indicate that when customers purchase beef, the likelihood of them purchasing whole milk also increases. For patterns with high confidence, supermarkets can drive sales of these combined items through targeted marketing strategies, such as through personalized recommendations to promote this potential propensity to purchase.

Taking both support and confidence into account, patterns that have both high support and high confidence should be considered a priority for marketing and inventory management, and focusing on marketing these items tend to yield better returns.

5) Performance of System Recommendations Derived from Collaborative Filtering, and Pattern Mining

RMSE of Collaborative Filtering Recommendations	**Accuracy of Pattern-Based Recommendation Method**
~ 2.35	**~ 2%**

For the collaborative filtering to produce a RMSE of around 2.35, signifies extremely low performance, as the user item matrix is highly sparse, where most users may have only bought 0 or 1 of each item. With the pattern-based system, although 2% accuracy is low as well, it can be reasoned that this is a slightly superior score to the collaborative filtering, due to the seeming sporadicity of the dataset, thus obtaining even such few correct recommendations can be seen as a win.

6) The system takes roughly 7.5 minutes to complete, on 38765 samples/transactions, with the bulk of the runtime being spent on the collaborative filtering. Thus, one can reasonably guess than when the problem size is increased by around 26 times, to 1 million transactions, the runtime also elongates by approximately the same order, to around 190 minutes (~3 hours). The system was run on a computer with an ‘AMD Ryzen 9 5900HX with Radeon Graphics’ processor.

Conclusion and Recommendations

As mentioned prior, although both recommendation methods tested were far from acceptable in performance, the itemset-based recommendations perhaps slightly edge its collaborative filtering alternative. Reasoning for this is that the collaborative filtering's RMSE, appears to be extremely unacceptable, whereas some sales-generating value can be seen from the other method's low accuracy.

On this relatively minor dataset, runtime was around 7.5 minutes, and predictions suggest around 180 minutes runtime when dealing with 1 million samples. As recommendations for users may need to be generated on the daily, this roughly 3-hour operation time is impractical, thus the system architecture should be reviewed to optimise efficiency.

With the lowly performance numbers, there is seemingly minimal benefit from this project, however, the systems and results obtained may be applicable for assisting the development and testing of future, more well thought through recommendation methods. Also, the mistakes in design and implementation of this project can serve as a guide for future work. Moreover, it may be of consideration to trial the itemset-based recommendation method on the store site and examine changes in sales or customer engagement thereafter. Additionally, the frequent item-sets determined can be utilised to re-structure a selection of stores, in a trial environment, to evaluate real-world impact.

To enhance future endeavours concerning this issue, it may be useful to diversify the customer details collected from the West Wing Supermarket Customer Points scheme, to improve data dimensionality. Also, the parameters used in the project for pattern mining and recommending, could be reviewed, and more extensive testing can be conducted for all possible options.

Reflections

Following the development of the collaborative filtering method, the importance of having a reference point for performance evaluation was further emphasised. This was as after the RMSE of the performance of the collaborative filtering system on the test set was calculated, the level of performance of the system was not clear as there was nothing appropriate to compare it to.

Thereafter, an idea for a hypothetical, sort of 'worst-case' RMSE was manufactured, which involved designing a dummy-system, that recommended for each user, every item they did not purchase with the 'mean' of the test user item matrix. This 'mean' is the average of all user-item counts within the test-set derived user item matrix. Thus, the RMSE would then be derived from the loss between this 'average', and the actual test set records. Having such a reference point may have slightly improved the results discussion.

Through this project, we learned about the true value of data and the application of various data mining techniques in the field of product recommendation. The historical customer purchase data seems to be very normal. In addition to simple plotting and calculations, we can mine this kind of data deeper to find out the potential relationships between products. These relationships are abstract and valuable. Moreover, we learned to apply pattern mining algorithms such as FP-growth. We might try to combine more methods together to provide more accurate results and optimize the structure of the system to reduce the running time in the future.

We have successfully identified common combinations of goods purchased using FP-growth algorithm. We believe that proper data cleaning and preprocessing are essential steps before applying any data mining algorithm. Through this project we have learned how to effectively process and prepare data to ensure that the algorithm works optimally. Also, in this project we have only analysed the goods in the same basket, but in future when we need to take more data into consideration, we will also be able to do better data analysis as well as data mining. Not only did we learn how to apply complex data mining techniques to solve real-world problems, but we also learned that continuous learning and improvement of algorithms is the key to adapting to the ever-changing needs of the market.

Contribution Appendices

Chunyu Zhang:

In the code of this project, I am responsible for the part of the code that performs frequent mining. The entire code for frequent mining was done by me. And my group member thought that interest is the best data for evaluation, and he needed a list with patterns and interest as input, so this part of the code was also done by me. In addition, I wrote code to output five examples of frequent patterns with their confidence and support on both training and as well as a table of metrics that showing results of testing frequent patterns on the test set.

In the report section, I was responsible for section 5, how to get the pattern results in section 8, what metrics to use to evaluate the patterns and how to rank the patterns. There is also a second question in part 8 that shows 5 examples, and a metrics chart with discussion in part 4. Finally, in Part 9, we decided on the methodology we wanted to recommend after the discussion, and I completed this part of the report.

Ebubekir Pulat:

In the report, I contributed to the 'Executive Summary', and the sections with the headings 'Introduction', 'Collaborative Filtering', 'Turning Frequent Item-Sets into Recommendations', 'Results', 'Conclusion and Recommendations' and 'Reflections'.

Regarding code, I contributed to the code that performed collaborative filtering, and generated recommendations from frequent item-sets.

Xingyuan Zhang:

In the report, I contributed the 'Introduction', 'Exploratory analysis', parts of 'Conclusion and Recommendations' and 'Reflections'. For the code, I did additional research on generating recommendations based on SVD algorithm and provide an example on how to find patterns by SVD decomposition.