A Utility-Based News Recommendation Engine

Student: Xing Zhao, Supervisor: Aijun An
EECS4080 Fall 2015
Electrical Engineering and Computer Science

1. Introduction

Since the Internet came to the history of mankind, tons of information have been generated from almost everywhere on the globe. To select articles among these mega news resources that interest users the most turns into a popular research topic. News recommendation systems are invented and deployed in many mainstream newspaper corporations such as Globe and Mail, CBC and the New York times. When you visit their website and open any of the headlines, a sidebar called “related coverage” or “more news” will appear on the right side of the page listing five to ten news headlines to attract you for further reading. This sidebar is driven by a news recommendation system which provides a list of news for a user to select based on the previous viewed news. In this project, we aim to develop a news recommendation system that provides the most wanted news for users upon their initial reading.

Different news recommendation systems have various features such as the feature of recommending news based on most viewed or content related articles. Content based method plays a key part in various domains such as E-commerce and news. In news domain, this method collects news articles based on the user’s interest and provides a personalized newspaper to the user. However, in order to improve the accuracy of the recommended articles, some content based technique [1] demands a user profile for each user. Thus, a user has to fill in one’s own profile and keep it up-to-date manually. This extract work becomes a burden to a user. Few companies are willing to employ it into their business. Frequent Pattern Mining (FPM) is one of the most prevalent algorithms implemented in news recommendation systems based on the pages views and clicks from users. Fu et al. [2] has developed a webpage recommender system using a classical FPM, Apriori algorithm to generate association rules over users’ browsing histories. Likewise, it is a vital supplement to the content based technique [3]. However, FPM only takes statistical information of each news article into count. The article that was most-frequently clicked-on by users is recommended by it. Therefore, it misses a critical attribute, the value of how much time the user spent on the article. A user may not finish reading the article depending on whether the user likes the news or not. Not to mention, it is common for a user mistakenly or accidently clicking on a news article. In these cases, FPM loses its accuracy on its recommended news articles. This limitation leads us to consider a more comprehensive mining algorithm that is able to include the time attribute. High utility itemset(HUI) mining [4] introduces a feature named utility which is used to describe the value of an item (a news article in this case). In HUI, the utility of an item comprises internal utility and external utility which provide a payload to carry the value of time duration and additional element such as the popularity of news. So we integrate HUI into our recommendation engine. Our engine not only includes the time spent by users per news but also takes on the popularity of a news item.

In this project, we define the utility for a news item and introduce a novel definition on a utility-based association rule mining. Thereupon, we use the mined high utility patterns to generate a list of utility-based association rules which are ultimately used for news recommendation. In the evaluation, we compare our approach with the Apriori technique [6] as well as a prior utility-based approach [7] under the same real dataset from Global and Mail and demonstrate the performance of our news recommendation engine.

2. Related Work

High utility pattern mining includes but not limited to sequential pattern mining and non-sequential pattern mining. In reality, the former usually offers better prediction than the latter in terms of the accuracy and the latter outperforms the former on personalized recommendation systems [5]. Nonetheless, sequential pattern mining requires detailed information about the items such as it uses timestamps for grouping the near items in duration into an itemset in a transaction. Manny datasets do not keep timestamps due to the privacy concern. Non-sequential pattern mining disregards the time sequence and therefore can accommodate general datasets under the privacy regulation. Thus, we adopt it into our approach.

Traditional association rule mining is originally designed for market basket transactions analysis such as Apriori association rule mining [6] which only describes the correlations between items in a transaction. Diaper → Beer only reflects that a pattern among transactions where {Diaper} is statically related to {Beer}. It does not necessarily distinguish the order of the items such as transactions {Diaper, Beer} and {Beer, Diaper} all support the previous pattern. However transactions like such do not share the same pattern in news recommendation systems.

Utility-based association rule mining is relatively new but has been studied by Sahoo et al. [7]. The definition of the utility confidence of a rule is the central piece of Sahoo's utility-based association rule mining. It emphasizes the proportion of the utility of the antecedent of a rule among the total utility of the same itemset(as the antecedent) in the dataset. For example, for a rule Diaper → Beer, the utility confidence of this rule is the total utility of Diaper from transactions that contain this pattern divides the total utility of Diaper in all transactions. The definition promotes a pattern having the antecedent with a high ratio between its utility in the pattern and its total utility of transactions. This approach does not directly benefit the news recommendation systems since the antecedents of rules are used to match the news that a user initially read, yet the consequents of rules are the crucial piece for recommending news. However, it gives rise to our definition on the utility confidence of a rule for news recommendation. Our definition promotes a pattern having the consequent with a high proportion of the pattern in utility. Besides, we emphasize the consequent of a rule must be a high utility itemset so that we can recommend news set having high consumption of reading time by users.

3. Design and Implementation

3.1 Define the utility of a news item

In our method, the utility of a news article consists of internal utility and external utility. Henceforth we use item to describe news article. The time duration a user spent on a news item is defined as the internal utility and the external utility uses the count of the shares of the item via social media or the users' rating on the item. If a user is willing to spend a relatively longer time on reading an item, the item is most likely more valuable to the user. If a group of users spend more time on a news item, the item has more value to the group. So do to a community of users. If the assumptions are true, another user from the same group or community will have better chance to enjoy the news that the majority members are mostly interested in. If a news item attracts more users to share, comment or rate, it is worth more social recognizable value. As such, we use the count of the shares of news on social media and or the number of the users' rating as the external utility. We define the utility of a news item the product of its internal and external utilities.

The use of the utility improves the accuracy of recommended news comparing to FPM based recommendation systems since a user may click on a news page and close it right after a glance or in a worse case, the page is accidently clicked by the user. The news item in such situations will only receive a fairly low utility in our approach.

The definitions of the calculation for news items are described as below:

Utility of an item *i* in a transaction ***T*** is defined as: $u(i,\ T_j)\ = q(i,\ T) \times p(i)$, where ***q(i, T)*** is the quantity of item ***i*** in transaction ***T*** (the internal utility) and ***p(i)*** is the external utility of item ***i***.

Utility of an itemset *X* in a transaction ***T*** of a dataset ***D*** is defined by:
$u(X,\ T_j) = \ \sum_{i \in X\ \cap\ T_j \in D} u(i,\ T_j)$.

3.2 Overview of the framework

The framework of our News Recommendation Engine is illustrated in Figure 1.

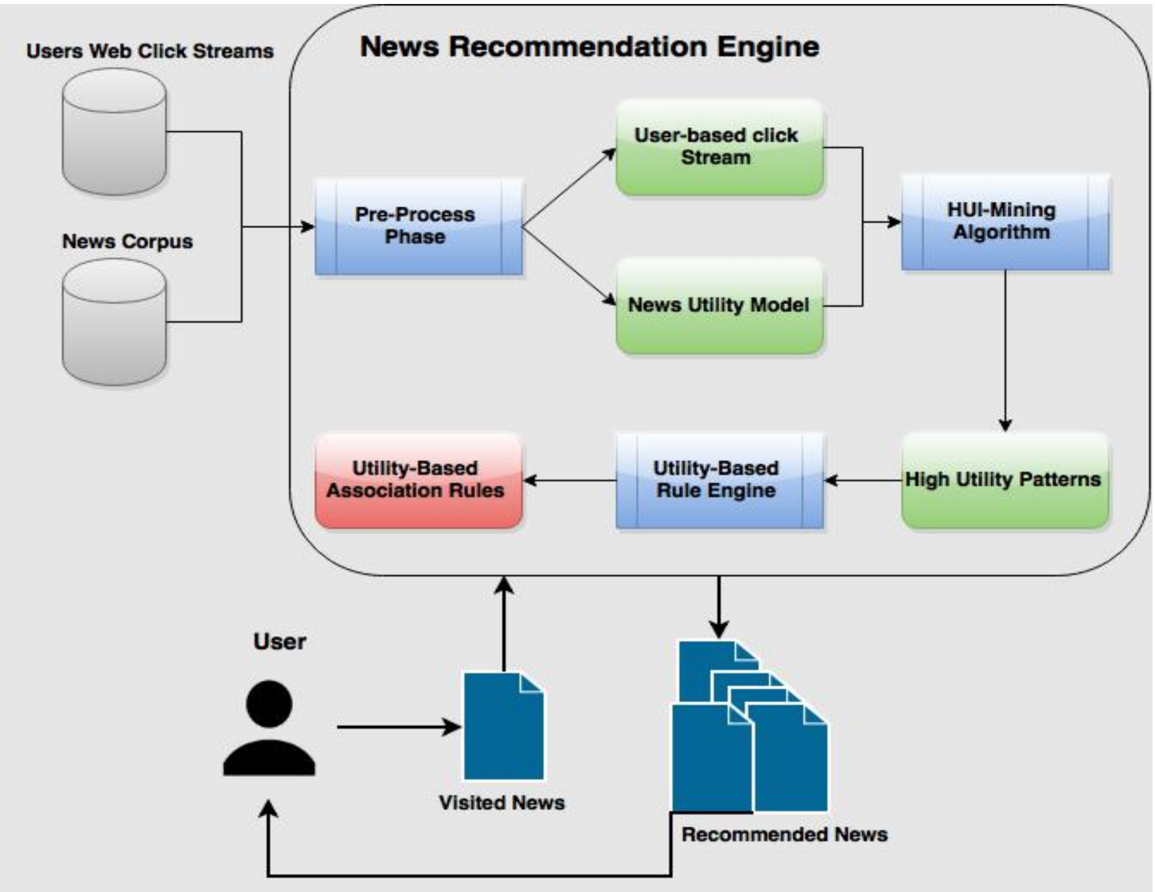


Figure 1. News Recommendation Engine Framework

At the top left corner of Figure 1, the processing flow starts with the raw dataset which includes "Users Web Click Streams" and "News Corpus". Then, it goes to "Pre-Process Phase" where the raw dataset is transformed into the required format of the component in the following stage. "User-based click Stream" and "News Utility Model" are the results of the "Pre-Process Phase". Before they are passed to the next stage, the internal utility and the external utility are calculated and combined into the item utility. "News Utility Model" provides an interface for defining and manipulating the internal and the external utilities. "HUI-Mining Algorithm component" runs on the processed dataset to find the "High Utility Patterns/Itemsets"(HUIs). Thereafter, "Utility-Based Rule Engine" runs on the HUIs and generates "Utility-Based Association Rules". Based on these generated rules, we recommend the news to the user. While the user is continuing browsing the news, the browsing events are sent back to the engine and added to the raw dataset, the component before the "Pre-Process Phase". As a next life cycle of the flow starts, the utilities of the events are explored and utilized. In this way, our news recommendation engine can follow users' reading selections and behavior patterns and keep itself up-to-date.

3.3 Pre-Process of the raw dataset

The raw dataset contains the news ID and time spent by the user per record. When the utility values are extracted from each entry line in the raw dataset, they need to be normalized. The time spent by users on the news is normalized by dividing the word count of the news content. Since some record indicates a user spent 9 hours or longer on the news which is abnormal, we consider in average a regular user normally spends 30 minutes on a news item. Any duration longer than that, we assume the user left the screen without closing the browser. Thus, any time spent longer than 30 minutes, we normalize it to 30 minutes and divide it by the word count. Due to the dataset has no valid transaction information for each record, we define a user's reading activity of a day as one transaction. In order to simulate a daily reading activity for a user, we assume a regular user read 5 news articles in average with a deviation of 2 per day, that is, randomly a user may read 3 to 7 news per day. Then, we partition the raw dataset into transactions with 3 to 7 items in random (in average of 5 and a deviation of 2). Meanwhile, for each transaction, the utility of every item is computed and mapped to the relative items by storing items and their utilities into two arrays respectively and orderly (refer to Figure 2). Since each item has its value and both share same index in arrays, we get a mapped item-value array.

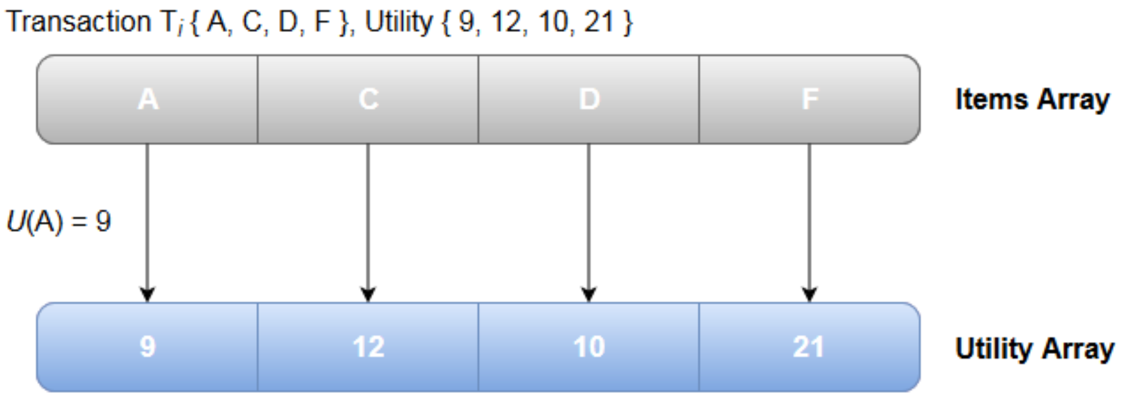


Figure 2. Items Array and its related Utility Array

3.4 HUIminer in HUI Mining Algorithm

We implement Liu's HUIminer algorithm [4] into our engine to find the *HUIs* from the dataset. Some changes are made in the implementation. We take advantage of the utility lists (ULs, refer to Figure 3) produced by HUIminer during the mining. We extract the utility of each item having the same transaction ID from the elements of the UL and add them up to get a local utility [7] for each item of a high utility itemset. Thereafter, every HUI contains a local utility array (or utility unit array [7]) that holds the local utility value of each item. HUIs along with their local utility array are used in the next stage, utility-based association rule mining.

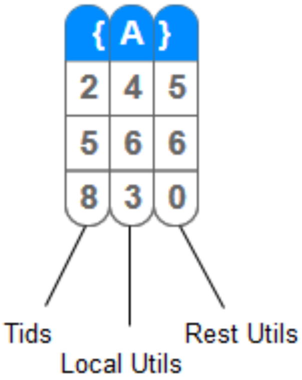


Figure 3. Initial Utility List for item {A}

We introduce the definitions to calculate the utility:

High Utility Itemset (*HUI*): An itemset ***X*** is called a high utility itemset (*HUI*) on a dataset ***D*** if and only if ***u(X)* ≥ *min_util*** where ***min_util*** is the threshold.

Local Utility Value (*luv*): The utility value of an item ***i*** in an itemset ***X***, is defined as ***luv(i, X)*** which is the sum of the utility values of the items ***i*** in all the transactions where ***X*** resides such as: $luv(i, X) = \sum_{X \subseteq T_j \cap T_j \in D} u(i, T_j)$

Utility Unit Array: For an itemset ***X***, its utility unit array ***U(X)*** stores the local utility values of each item of ***X***.

3.5 Utility-based Association Rule mining

In order to utilize the HUIs for the recommendation, we define utility-based association rule mining to parse each HUI. Our definition is different from prior utility-based association rules mining algorithms in terms of how the confidence is defined.

We introduce a new utility-based association rules mining as follows:

Utility-Based Association Rules: Given two itemsets X and Y, $X \rightarrow Y$ (X $implies$ Y) is an association rule R ***iff***:

1) X $\neq \emptyset$, Y $\neq \emptyset$ and X $\cap$ Y $= \emptyset$.

2) Y and $X \cup Y$ are high utility itemsets.

3) Its **utility confidence** (defined as ***uconf(R)*** = $\frac{luv(Y,\ X \cup Y)}{u(X \cup Y)}$) is no less than a user-defined minimum utility confidence (***min_uconf***).

The utility confidence reflects how much weight the recommended news covers in the total utility of a rule. A higher utility confidence indicates a higher total utility value of the recommended news in a pattern. When all generated rules are sorted in descending order by the utility confidence, the news items with the most utility (time spent by users and popularity) from rules containing same items are recommended.

Utility-Based Association Rule Mining Algorithm:
Input: a list of mined high utility itemset from HUIminer
Output: a list of utility-based association rules

```
1.  for each HUI mined from the dataset:
2.     X, a HUI is segmented into two itemsets, X:{antecedent → consequent},
          generate a combination of subsets X:{antecedent → consequent}, each with
          unique items
3.     for each combination of X:{antecedent → consequent}
4.        the utilities of both are compared with min_util respectively
                /* the local utility of antecedent, luv(antec) */
                /* the local utility of consequent, luv(conseq) */
5.           if either the antecedent or the consequent is HUI,
6.              the uconf is compared with min_uconf (a specified threshold)
7.              rule “antecedent → consequent” is added
                   iff $uconf = \frac{luv(conseq)}{u(X)} \geqslant min\_uconf$  /* the utility of X, u(X) */
8.              rule “consequent → antecedent” is added
                   iff $uconf = \frac{luv(antec)}{u(X)} \geqslant min\_uconf$
```

Note that in line 7 and 8, we use the two way comparison to find a rule by the uniqueness of each subset. In line 2, we generate the subset with the size of one (item) up to the floor of the half size of the HUI under the two way comparison. Figure 4 illustrates the two way comparison and the uniqueness of the subsets on a HUI.

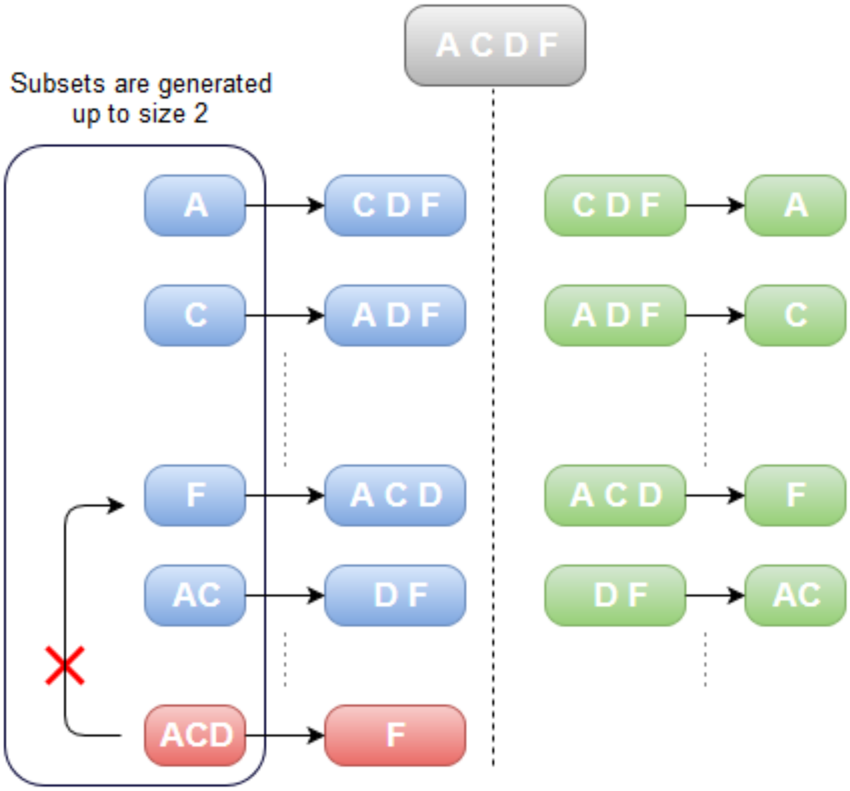


Figure 4. HUI {ACDF} is segmented into antecedent and consequent

Using utility-based association rule mining, sample rules are generated as below:

[The self-serving case for expanding Billy Bishop] ⇒ [Toronto's construction: why is it happening all at once?] [Quebec's construction firms battle wary clients as Charbonneau unfolds] [Toronto committee delays debate on controversial island airport expansion plans] [Toronto's island airport: What's at stake?] [Daycare demand soaring in Toronto region as YMCA adds more spaces]

[La Prairie, Quebec mayor dies from wasp stings] ⇒ [Suspect appears in court as search for Calgary family continues] [I know from experience the horror of airliner attacks] [A fund for the fearful, and fearful of missing out] [All my friends have married off. How do I attract new ones?] [The man who foresaw the 2008 crash now says stocks are in a highly advanced bubble] [The big hole in Canada's public health leadership needs to be filled]

[Was I a wanted man in the Donetsk Peoples Republic?] ⇒ [Is it time to increase maximum speed limits?] [Video: Ride-along for teen turns into police car chase] [Suspect appears in court as search for Calgary family continues] [Anthony Bennett, Andrew Wiggins offer a preview for upcoming season] [MH17: Downing of plane pushes Russia and the West into deeper confrontation]

4. Evaluation

Evaluation measures are used to provide metrics such as precision and recall for revealing the performance of a recommendation system. Conventional evaluation measures calculate the precision and the recall based on the occurrence of each item. It might be biased towards the Frequent Pattern Mining using the statistical counts of items. Utility-based association rules are relatively new while evaluation measures that take the item's utility into account are rare at the moment. In this project, we integrate the utility value into the conventional evaluation measures and implement a utility-based evaluation measures. The results from both evaluation methods are compared and analyzed.

The following table (refer to Figure 5) and equations illustrate how the precision and the recall are defined and calculated by conventional evaluation measures.

		Reality	
		Actual Good	**Actual Bad**
Prediction	**Predicted Good**	**True Positive(TP)**	**False Positive(FP)**
	Predicted Bad	**False Negative(FN)**	**True Negative(TN)**

All recommended Items

All good items

Figure 5. Evaluation on recommendation systems

Metrics:

$$\boldsymbol{Precision} = \frac{TP}{TP + FP} = \frac{|Good\ items\ recommended|}{|All\ recommendations|}$$

$$\boldsymbol{Recall} = \frac{TP}{TP + FN} = \frac{|Good\ items\ recommended|}{|All\ good\ items|}$$

$$\boldsymbol{F\text{-}measure} = 2 \cdot \frac{Percision \cdot Recall}{Percision + Recall}$$

In our evaluation, precision is the intersection of the recommended items and the actually reviewed items divided by the recommended items in utility. Recall is the intersection of the recommended items and the actually reviewed items divided by the actually viewed items in utility. The occurrence evaluation then calculates occurrence counts instead of utility.

Having integrated the utility into the calculation, we introduce the following definitions:

Utility of good (viewed) recommended items from a rule r :{ ***X***→***Y*** **}** in a transaction *T* of a dataset *D* is defined by:

$$u(r,\ T_j) = \begin{cases} \sum_{i \in Y \cap T_j \in D} u(i,\ Y) & for\ precision \\ \sum_{i \in Y \cap T_j \in D} u(i,\ T_j) & for\ recall \end{cases}$$

Precision in a transaction (*pit*) [8]: The utility value of the intersection of the recommended items *R* for this transaction *T* and the actually good (viewed) items *G* in *T* divided by the utility value of *R* for *T* such as:

$$pit = \frac{u(R,\ T_j) \cap u(G, T_j)}{u(R, T_j)} = \frac{u(r, T_j)}{u(R, T_j)}$$

Recall in a transaction (*rit*): The utility value of the intersection of the recommended items *R* for this transaction *T* and the actually good (viewed) items *G* in *T* divided by the utility value of *G* for *T* such as:

$$rit = \frac{u(R,\ T_j) \cap u(G, T_j)}{u(G, T_j)} = \frac{u(r, T_j)}{u(G, T_j)}$$

Precision for a dataset *D*: The average of the *pit*s of all transactions in *D*, each *pit* is the average of possible combinations of *pit* per transaction (e.g. transaction *i* <Ti: BCD> has combination of {B→CD} and {BC→D}).

$pd = \frac{\sum_1^n pit_{j \cap T_j \in D}}{n}$ where $pit_{j \cap T_j \in D} = \frac{\sum_1^k pit_{j \cap T_j \in D}}{k}$ and *n* is the number of translations in *D* ,*k* is the possible combinations of each transaction.

Recall for a dataset *D*: The average of the *rit*s of all transactions in *D*, each *rit* is the average of possible combinations of *rit* per transaction.

$rd = \frac{\sum_1^n rit_{j \cap T_j \in D}}{n}$ where $rit_{j \cap T_j \in D} = \frac{\sum_1^k rit_{j \cap T_j \in D}}{k}$ and *n* is the number of translations in *D* ,*k* is the possible combination of each transaction.

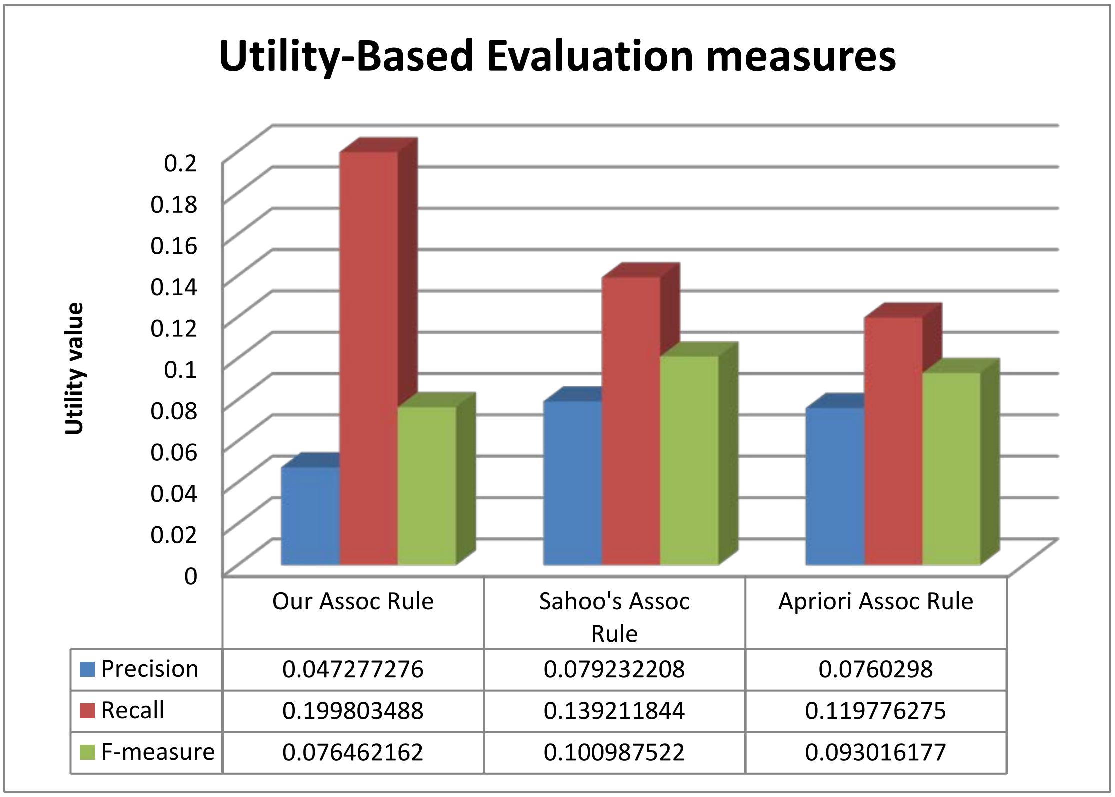
The evaluation procedures:

1. Partition the dataset into Training set and Test set by 7:3 ratios in terms of the transactions.
2. Run the News Recommendation Engine on the Training set to obtain a list of rules and the local utility array per rule.
3. The list of rules is aggregated based on the antecedent. If antecedents of rules are same, then their consequents are merged so that each item in the consequent of the new aggregated rule is unique. (e.g. A→BC, A→BD is aggregated to A→BCD)
4. With a list of aggregated rule, we start the evaluation on the Test set.

```
5. for each transaction of the Test set
6.     for each item in the transaction exclude last item
7.         if any combination of this item and its following items in order exists
                                                     in any antecedent of rules
8.               Precision and Recall are computed for this combination
       endfor
9.     the average Precision and the average Recall are computed in this transaction
   endfor
```

10. the average Precision and the average Recall are computed for the Test set

To better compare our utility-based association rule mining with the frequent association rule mining, we introduce a classical FPM algorithm, Apriori association rule mining [5] into the evaluation. We also include the Sahoo’s utility-based association rule mining [7] for peer comparison. Having run the utility-based evaluation method on the test set with all three algorithms, the results are shown in Figure 6.

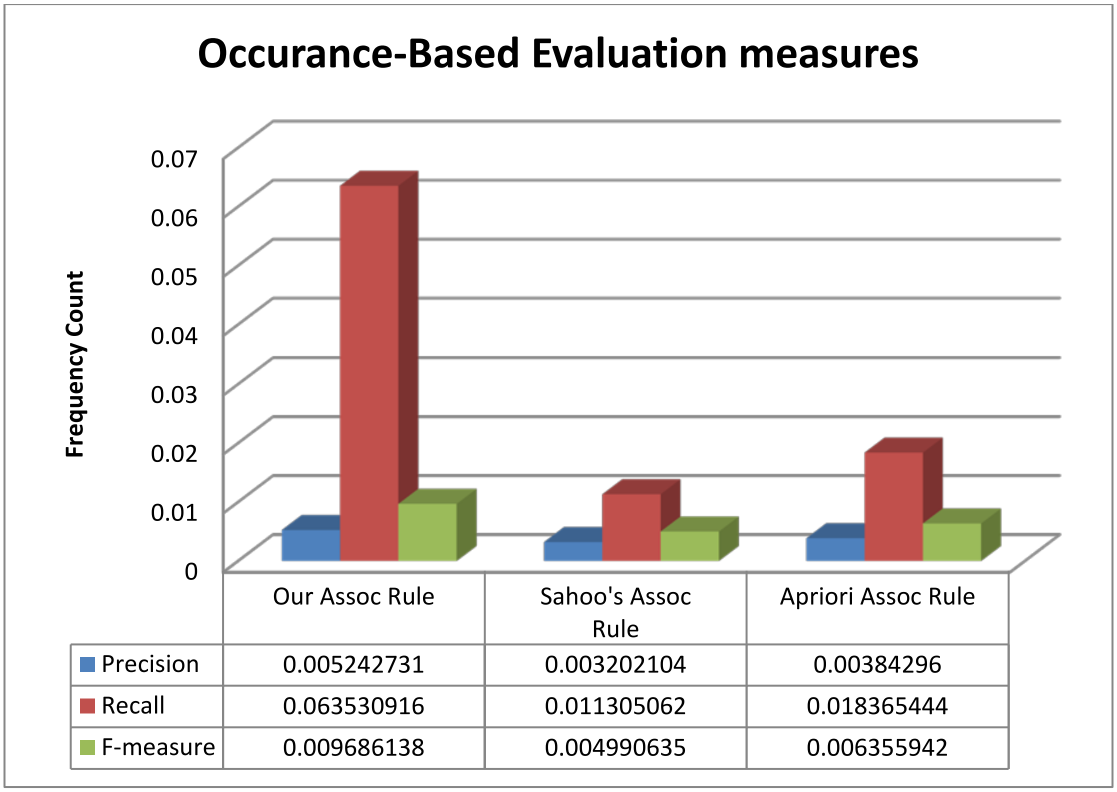


	Our Assoc Rule	Sahoo's Assoc Rule	Apriori Assoc Rule
🟦 Precision	0.047277276	0.079232208	0.0760298
🟥 Recall	0.199803488	0.139211844	0.119776275
🟩 F-measure	0.076462162	0.100987522	0.093016177

Figure 6. Utility-Based Evaluation results

The recall of our utility-based association rule mining is 0.08 higher than the Apriori association rule mining (Apriori) and 0.06 higher than the peer algorithm based on the results. But its precision is 0.03 lower than Apriori and Sahoo’s approach. The recalls indicate the utility of the recommended news that are actually viewed by users from our rules is higher than the ones from Apriori. The reason is while the actually viewed news by the user (the divisor) is same for both mining algorithms in the recall calculation; our rules recommend news with more time values. The precisions indicate the utility of actually viewed news by the user has more weight among the recommended news by Apriori than by our rules. Note that the utility of the recommended items from each algorithm, the divisor is different in the precision calculation. From the training dataset, Apriori mostly generates 1-itemset rule meaning there is only one item for recommendation from a generated rule. In the case that for a transaction Apriori has only one item to recommend and our approach has two items to recommend. When both have one item viewed by the user, Apriori receives 100% precision which is definitely higher than our approach regardless how much utility of the news recommended.

We also conduct the conventional evaluation (refer to Figure 7) to get a comparison among all three algorithms using the occurrence.



	Our Assoc Rule	Sahoo's Assoc Rule	Apriori Assoc Rule
🟦 Precision	0.005242731	0.003202104	0.00384296
🟥 Recall	0.063530916	0.011305062	0.018365444
🟩 F-measure	0.009686138	0.004990635	0.006355942

Figure 7. Occurrence-Based Evaluation results

The result shows that our approach outperforms the others on the statistical count of the occurrence of the recommended items. That implies our approach offers more accurate predictions in precision and better sensitivity than its rivals in recall.

Lastly, we compare the rules generated from our utility-based association rule mining and Apriori association rule mining and the time utility of the recommended news by each rule. The following tables list the top 3 rules generated from both algorithms.

Top 3 Utility-Based Association Rule by utility confidence	**Time (minutes)** news recommended
[Maple Leafs avoid arbitration with Cody Franson] ==> [Seven Days of Television: July 28 to August 3] [As trade rumours swirl, Toronto eyes first place] [Can the Jays make a playoff push without mortgaging the future?] [Seven Days of Television: August 4 to August 10] [Another typical day in Hogtown]	1951

[Meet the man who's driving Argentina to the brink] ==> [Ontario is left with few options to fix its revenue problem] [Unable to reach deal with creditors, Argentina goes into default] [Sorry, Mr. Obama, this is a new Cold War. Here are seven reasons why] [Carrick on money: The four numbers that tell if you're financially healthy] [S&P revises outlook on Canada's big six banks to negative]	1783
[Friday's analyst upgrades and downgrades] ==> [Five surefire ways to maximize your life, starting this morning] [In Sierra Leone, we've stopped shaking hands] [Canadian couple detained in China did charity work in North Korea] [Ex-premier Redford resigns as Alberta MLA] [Olivia Wilde breastfeeds son in Glamour photo shoot] [Nigeria declares Ebola emergency; WHO says outbreak 'will get worse']	1722

Top 3 Apriori Association Rule by confidence	**Time (minutes)** news recommended
[Andrew Hallam's model indexing portfolio] [Chris Umiastowski's model growth portfolio] ==> [Norman Rothery's model value portfolio]	0
[Interactive Map: Path of Malaysia Airlines Flight MH17] [One Canadian among 298 dead in plane crash: reports] ==> [MH17: Disaster ratchets up Russia-Ukraine tensions]	0
[In photos: The Ford brothers and their Etobicoke connections] ==> [Globe investigation: The Ford familyï's history with drug dealing]	4270

According to the above tables, the time of the recommended news per rule by our approach is 395 minutes more than by Apriori in average. Note that the top 2 rules from Apriori recommend 0 minute news which demonstrates that Apriori only has the mind of the market basket transactions analysis but does not fit with the news recommendation since it only tells the statically correlation between items based on the occurrence and is blind to the correlation and the order between items with their utilities.

5. Conclusion

In this project, we define and implement the utility-based news recommendation engine based on the high utility itemset mining and our defined utility-based association rule mining. The engine is able to recommend the most up-to-date interesting news (most time spent by users) to a user according to one's pervious reading.

Our utility-based association rules are the final products of our news recommendation engine at this point. Further work involves generalizing the news rules (relations) into topics relations by generalizing news IDs into the topic IDs using a mapping file. The mapping file contains entries of news ID and its related topic ID in a proper probability degree. As news rarely stays hot for a long period, once it loses its freshness, general topic relations or rules would be more useful and practical for news recommendation.

References:

[1] Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web* (pp. 325-341). Springer Berlin Heidelberg.

[2] Fu, X., Budzik, J., & Hammond, K. J. (2000, January). Mining navigation history for recommendation. In *Proceedings of the 5th international conference on Intelligent user interfaces* (pp. 106-112). ACM.

[3] Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 31-40). ACM.

[4] Liu, M., & Qu, J. (2012, October). Mining high utility itemsets without candidate generation. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 55-64). ACM.

[5] Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on* (pp. 669-672). IEEE.

[6] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[7] Sahoo, J., Das, A. K., & Goswami, A. (2015). An efficient approach for mining association rules from high utility itemsets. Expert Systems with Applications, 42(13), 5754-5778

[8] Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2001, November). Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd international workshop on Web information and data management* (pp. 9-15). ACM.