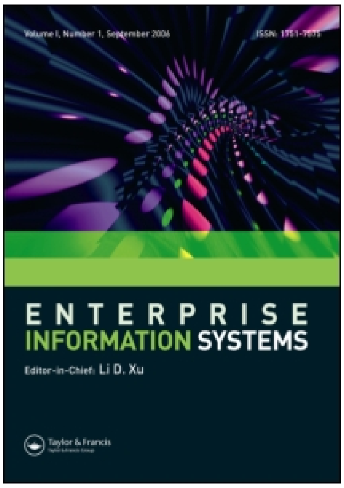


**Enterprise Information Systems**
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t748254467

# Healthcare information systems: data mining methods in the creation of a clinical recommender system

L. Duan[a]; W. N. Street[a]; E. Xu[bc]
[a] Department of Management Sciences, Henry B. Tippie College of Business, University of Iowa, Iowa City, IA, USA [b] College of Natural Resources, University of California-Berkeley, Berkeley, CA, USA [c] College of Arts and Sciences, University of Virginia, Charlottesville, VA, USA

First published on: 20 January 2011

**To cite this Article** Duan, L. , Street, W. N. and Xu, E.(2011) 'Healthcare information systems: data mining methods in the creation of a clinical recommender system', Enterprise Information Systems, 5: 2, 169 — 181, First published on: 20 January 2011 (iFirst)

**To link to this Article: DOI:** 10.1080/17517575.2010.541287

**URL:** http://dx.doi.org/10.1080/17517575.2010.541287

*Enterprise Information Systems*
Vol. 5, No. 2, May 2011, 169–181



# Healthcare information systems: data mining methods in the creation of a clinical recommender system

L. Duan[a]*, W.N. Street[a] and E. Xu[b,c]

[a]*Department of Management Sciences, Henry B. Tippie College of Business, University of Iowa, Iowa City, IA 52242, USA;* [b]*College of Natural Resources, University of California-Berkeley, Berkeley, CA 94720, USA;* [c]*College of Arts and Sciences, University of Virginia, Charlottesville, VA 22904, USA*

(*Received 15 November 2009; final version received 15 November 2010*)



Recommender systems have been extensively studied to present items, such as movies, music and books that are likely of interest to the user. Researchers have indicated that integrated medical information systems are becoming an essential part of the modern healthcare systems. Such systems have evolved to an integrated enterprise-wide system. In particular, such systems are considered as a type of enterprise information systems or ERP system addressing healthcare industry sector needs. As part of efforts, nursing care plan recommender systems can provide clinical decision support, nursing education, clinical quality control, and serve as a complement to existing practice guidelines. We propose to use correlations among nursing diagnoses, outcomes and interventions to create a recommender system for constructing nursing care plans. In the current study, we used nursing diagnosis data to develop the methodology. Our system utilises a prefix-tree structure common in itemset mining to construct a ranked list of suggested care plan items based on previously-entered items. Unlike common commercial systems, our system makes sequential recommendations based on user interaction, modifying a ranked list of suggested items at each step in care plan construction. We rank items based on traditional association-rule measures such as support and confidence, as well as a novel measure that anticipates which selections might improve the quality of future rankings. Since the multi-step nature of our recommendations presents problems for traditional evaluation measures, we also present a new evaluation method based on average ranking position and use it to test the effectiveness of different recommendation strategies.

**Keywords:** nursing care plan; recommender system; data mining; correlation; information value; medical informatics; healthcare integrated information systems; healthcare enterprise-wide systems

## 1. Introduction

According to a report published in 2000 by the Institute of Medicine, at least 44,000 and perhaps as many as 98,000 patients die in the hospital each year as a result of medical errors alone (Iglesias *et al.* 2003). These data point to adverse healthcare events as the leading cause of deaths in the USA. Adverse events are estimated to

*Corresponding author. Email: lian-duan@uiowa.edu

ISSN 1751-7575 print/ISSN 1751-7583 online

DOI: 10.1080/17517575.2010.541287
http://www.informaworld.com

cost the nation between $37.6 billion and $50 billion; furthermore, preventable adverse events cost the nation between $17 billion and $29 billion (Iglesias *et al.* 2003). Patient care phenomena are so complex that it is difficult for many nurses to create effective comprehensive care plans for their patients (Bellika and Hartvigsen 2005). Three standardised nursing terminologies commonly seen in US nursing care plans are nursing diagnoses, encoded using NANDA (Nanda 2005); interventions, using NIC (Dochterman and Bulechek 2003); and outcomes, using NOC (Moorhead *et al.* 2005). Diagnoses are the identifiable problem, which we must rectify through intervention. The ultimate goal is to achieve an outcome tailor to the aforementioned diagnoses. The ultimate goal here is to interactively provide a ranking list of the suggested items in order to maximise efficiency and care quality in a hospital setting.

Researchers have indicated that integrated medical information systems are becoming an essential part of the modern healthcare systems. Such systems have evolved to an integrated enterprise-wide system. (Li and Xu 1991, Li *et al.* 2008, Yoo *et al.* 2008, Puustjarvi and Puustjarvi 2010). In particular, such systems are considered as a type of enterprise information systems or ERP system addressing healthcare industry sector needs (MacKinnon and Wasserman 2009). As part of efforts, our system simplifies the task of creating a comprehensive care plan for nurses by using previous input to suggest a course of action (Hardiker *et al.* 2002). For example, if a nurse has selected 'health maintenance' and 'pain acute', then the following list (Table 1) will appear. It shows factors that the nurse should consider in creating a comprehensive care plan. To contribute to the effectiveness, safety and efficiency of nursing care, we propose a nursing care plan recommender system. This system can facilitate clinical decision support nursing education, clinical quality control and serve as a complement to existing practice guidelines (Xu 1994).

Recommender systems have become an important research area since the appearance of collaborative filtering in the mid-1990s (Resnick *et al.* 1994, Hill *et al.* 1995, Shardan and Maes 1995). The interest in this problem-rich area is high because this research has a myriad of practical applications (Adomavicius and Tuzhilin 2005) that help users deal with a plethora of information by providing personalised recommendations, content, and services in compact lists. This allows users to waste less time by eliminating the need to search through endless lists of materials. Examples of such applications range from lists recommending books to CDs. A few examples of these specially tailored lists are products on Amazon.com (Linden *et al.* 2003), movies by Netflix (Koren 2008) and MovieLens (Miller *et al.* 2003), and news at VERSIFI Technologies (Billsus *et al.* 2002).

Table 1. A sample ranking list.

| Previous selection: | You have selected 28 (health maintenance), 12 (pain acute). | | | |
|---|---|---|---|---|
| Ranking list | Ranking | Code | Description | Value |
| | 1 | 52 | Knowledge deficit | 0.91 |
| | 2 | 37 | Risk for infection | 0.66 |
| | 3 | 39 | High risk for injury | 0.33 |
| | 4 | 68 | Physical mobility alteration | 0.19 |
| | 5 | 05 | Anxiety | 0.17 |
| | 6 | 78 | Skin integrity, impaired | 0.16 |
| | 7 | 67 | Self-care deficit, bathing/hygiene | 0.10 |
| | 8 | 79 | Skin integrity, risk for impaired | 0.05 |

Current recommender systems focus on commercial activities; thus, there are some differences from clinical activities. In clinical activities, nurses select all the required items for each care plan; however, in commercial activities, customers only select a number of the desired items in each transaction. Commercial recommender systems are not required to recommend all the desired items to customers; on the other hand, clinical recommender systems must recommend all the required items to nurses. Another factor separating commercial applications is that purchase behaviour is unary instead of binary. If a customer does not buy a particular item, it does not necessarily suggest that the customer dislikes it. The relationship between similar customers and a given item can be used to extrapolate the relationship between the customer and that item. In clinical recommender systems, this problem is not an issue because clinical behaviour is binary. Last, in commercial recommender systems there is a rating system, i.e. a scale from 1 to 5, illustrating how much the customer likes a particular item. In clinical recommender systems, there is rating system because a patient's requirement for a particular item is based on objective means and not on subjective desires.

It is our belief that the application of recommender technology to clinical nursing practice is relatively cutting edge, although there are several examples in literature regarding nursing expert systems (Ryan 1985, Keenan *et al.* 2006). Clinical expert systems are constructed according to the knowledge of experienced nurses, which creates a development bottleneck. This inevitably means as patterns change across time, the encoded rules need to be updated manually (Kakousis *et al.* 2010). By using data mining methods, we can extract rules from historical data automatically instead of relying on expert knowledge (Luo *et al.* 2007). These data mining measures are also capable of handling changes to practice standards by extracting patterns within sliding windows. Furthermore, data mining methods can deduce unique patterns for each individual hospital; thus, this is a far more accurate means for clinical quality control.

The article is organised as follows. The related work, focusing on collaborative filtering techniques, is presented in Section 2. The methodology and data structure we use is presented in Section 3. In section 4, we conduct a series of experiments to evaluate different methods. Section 5 concludes the article with an overall summary and possible directions for related future research.



## 2. Related work

In the most common formulation, the recommendation problem simply provides ranking list for items that a user has not encountered so far. With books and movies, recommender systems compile ranking lists by estimating ratings. Intuitively, this estimation is based on a given user's ratings for other items in a similar genre, on other users' ratings for a given item, or on other contextual information. Once we can estimate ratings for these unrated items, we can recommend to the user the item(s) with the highest estimated rating(s). More formally, the recommendation problem can be formulated as follows: Let $U$ be the set of all users and let $S$ be the set of all possible items that can be recommended. Let $p$ be a utility function that measures the usefulness of item $i$ to user $a$, i.e. $p$: $U \times S \rightarrow P$. Then, for each user $a \in U$, we recommend the item $i \in S$ that has the maximal user utility. Usually, the utility of an item is represented by a rating, which indicates how a particular user likes a particular item. However, depending on the application, the users can specify the utility $p$ by taking other criteria into account. Once the unknown ratings are estimated, actual recommendations of the $N$ best items to a user are made by selecting the $N$ highest ratings among all the estimated ratings for that user. The estimated ratings can be calculated in many different methods from machine learning, approximation theory and various heuristics. Recommender systems are usually classified into the following categories, based on how recommendations are made (Balabanović and Shoham 1997): content-based recommendations and collaborated recommendations.

In content-based methods, the utility $p(a,i)$ is estimated based on the utilities $p(a,i')$ in which the item $i'$ is similar to the item $i$. For example, in order to recommend movies to a certain user $a$, the content-based method tries to understand the commonalities, the profile of user, among the movies user $a$ has rated highly in the past, such as specific actors, directors, genres, etc. After this analysis, only the movies that have a high degree of similarity to the user's preferences would be recommended. The profiling information can be elicited from users explicitly through questionnaires, or implicitly from their transactional behaviours. Unlike content-based recommendation methods, collaborative recommender systems try to predict the utility of items for a particular user based on the user's rating for similar items (item-based), the ratings of this item given by other users (user-based) or through some models (model-based). User-based methods associate to each user its set of nearest neighbours, and then predict a user's rating on an item using the ratings of its nearest neighbours on that item. Item-based methods associate to each item its set of nearest neighbours, and then predict a user's rating on an item using the rating of the user on the nearest neighbours of the item considered. Since predicting the rating of a given user on a given item requires the computation of similarity between the user and all its neighbours that have already rated the given item, its execution time may be long for huge datasets. In order to reduce execution time, model-based approaches have been proposed. Model-based methods construct a set of user groups, and then predict a user's rating on an item using the ratings of the members of its group on that item. In many cases, different numbers of clusters are tested, and the one that led to the lowest error rate in cross-validation is kept.

Many evaluation measures (Herlocker *et al.* 2004) can be used to compare the results of different collaborative filtering methods. Given $T=\{(u, i, r)\}$ the set of (*user, item,rating*) triplets used for test, the most widely used evaluation measures are: (1) mean absolute error: $\mathrm{MAE} = \frac{1}{|T|}\sum_{(u,i,r)\in T}|p_{ui} - r|$; (2) root mean squared error: $\mathrm{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(u,i,r)\in T}(p_{ui} - r)^2}$; (3) precision: Precision $= N_{\text{desired\&retrieved}}/N_{\text{retrieved}}$; (4) Recall $= N_{\text{desired\&retrieved}}/N_{\text{desired}}$. The first two are used to measure how close the predicted rating is to the actual rating. The third one is used to measure how useful the top-$N$ ranking list is, and the fourth one measures how many useful items are retrieved. However, evaluating recommender systems is inherently difficult for several reasons. First, different algorithms may be better or worse on different data sets. Second, the goal of different evaluations is different. Third, it is hard to have one evaluation method to optimise multiple criteria when we have several goals. For example, the customer hopes to find a movie which is enjoyable, has a cheap price, and does not last too long. Accuracy and minimal error are not the ultimate goal (Herlocker *et al.* 2004, McNee *et al.* 2006). Some new algorithms appear to do better than the older algorithms, but all the algorithms are reaching a 'magic barrier' where natural variability prevents us from getting much more accurate. Hill *et al.* (1995) showed that users provide inconsistent ratings when asked to rate the same movie at different times. An algorithm cannot be more accurate than the variance in a user's ratings for the same item. A good recommender system should provide usefulness, not just accuracy. More often than not, minimal error leads recommender systems to the recommendation list containing similar items. Accuracy metrics cannot solve this problem because they are designed to judge the accuracy of individual item predictions; they do not judge the contents of entire recommendation lists. The recommendation list should be judged for its usefulness as a complete entity, not just as a collection of individual items. In addition to considering a recommendation list as a complete entity, under some circumstances, recommendations should also be considered as a sequence of actions, not isolated actions. We need to balance the gain between current and future recommendation lists. There is also an argument about whether or not a recommender should optimise to produce only 'useful' recommendations (for example recommendations for items that the user does not already know about). Recommending the item the user has already experienced does not provide any useful information; however, it does increase user faith in the recommender system.

To interpret recommendations we consider two dimensions: strength and confidence. More specifically, the strength of the recommendation asks how much the user likes the item. The confidence of the recommendation asks how sure we are that the recommendation is accurate. Many recommender systems conflate these two dimensions inaccurately. They assume that a user is more likely to prefer an item of five stars than an item of four stars. However, if a user is more likely to give four stars to item B than he or she is to give five stars to item A, it would be safer to recommend B instead of A. However, different short-term goals can lead to different preferences for types of recommendations. For instance, if a user wants to take his girlfriend to see a good movie, he might prefer the reliable four-star movie. If he wants to find a wonderful movie to watch alone and is willing to risk not liking the movie at all, it is good to select the less reliable five-star movie. To help users make effective decisions based on recommendations, recommender systems must help users navigate along both the strength and confidence dimensions simultaneously. Measuring the quality of confidence in a system is difficult since confidence itself is complex. When confidence is considered, how can we balance strength and confidence to provide better recommendations?

In summary, accuracy alone does not guarantee users an effective and satisfying experience. Instead, systems should be able to help users complete their tasks. A fundamental point proceeds from this basis: to do an effective user evaluation of a recommender system, researchers must clearly define the tasks the system is intended to support.

## 3. Methodology

### *3.1. Utility measurements*

As we mentioned in Section 2, we have two dimensions to interpret recommendations: the *strength* and the *confidence* of the recommendation. Most commercial recommender systems provide the ranking list according to the strength dimension. The most popular search engine, Google, does similar things: providing the ranking list according to how well the webpage matches the keyword(s). However, there are some examples in commercial applications on the confidence dimension. When a customer buys a book from Amazon.com, the website also recommends other books that customers have purchased together. Some low-rating books will be recommended before high-rating books. That part is related to finding the frequent itemsets, which contains the current book the customer wants to buy and other books, among transactions. Back to our clinical problems, there is no rating system on a scale from 1 to 5. Patients either need or do not need the item. Therefore, we can simplify the strength dimension and focus on the rarely exploited confidence dimension due to the binary nature of clinical behaviours. To facilitate electronic health record input, we provide a list of all possible selections. In each step, the nurse selects one required item from the list. Ideally, the item at the top of the list will be selected; thus, in general we wish to rank-order the list such that the selected items are as close to the top as possible. After each selection, the selected item is removed from the ranking list, and the list is re-ordered. Here, we use the commonly used measurements for association rules, such as support, confidence and lift (Han 2005), to construct the ranking list. In addition, due to the step-by-step process, we use a novel measure that anticipates which selections might improve the quality of future rankings. Throughout the rest of the article, we use $N$ to denote the total number of care plans. The notation $N(S)$ is used to denote the number of care plans which contain the itemset $S$.

The first measurement is support, the percentage of records in which the item appears. We use support to measure popularity and recommend the most popular selection to the user first. The support of a given item $A$ is calculated as $N(A)/N$.

The second measurement is confidence, the probability of the item being chosen conditioned on the previous set of selected items. The confidence of a given item $A$, given the set $S$ that has already been chosen, is calculated as $N(S \cap A)/N(S)$.

The third measurement is lift, the ratio of the item's confidence to its support. Hence lift gives us information about the increase/decrease in probability of the item being chosen given the previous set of selected items. The lift of a given item $A$, given the set $S$ that has already been chosen, is calculated as $confidence(A|S)/support(A)$.

We also introduce a new measure termed information value or simply $IV$. To measure $IV\,(A)$ we consider how 'orderly' the list of conditional probabilities would be if $A$ is chosen, and for that we use a variation of the entropy equation from information theory. Here, $p_i$ is used to denote the confidence of the $i$th remaining selection after if $A$ has been selected. The entropy for item $A$ is calculated as $\sum_{i=1}^{k} (p_i * log_2(p_i) + (1 - p_i) * log_2(1 - p_i))/k$. Ideally, any $p_i$ should be either 1 or 0, leading to an entropy of 0. In this case, we would be able to identify exactly the set of selections that must be chosen, given the current set of selections plus $A$. Conversely, the most undesirable case is a $p_i$ of 0.5. In this case, we have no information about future selections, and the future ranking list would be chaotic. We promote the selection that has both the high probability to be chosen and low entropy to predict future selections. With this measurement, we strike a balance between the gain of the current selection and that of future selections. The information value of the possible selection $A$ is calculated as $confidence(A|S) * (1 - entropy(A|S))$.

### 3.2. Data structure

Regardless of the measurement used, the fundamental element of this system is to easily obtain the occurrence of any selection set. Getting the occurrence of a set relies on a top-down search in the subset lattice of the items. Here, we use a prefix tree structure (Borgelt 2003) to quickly retrieve the occurrence of any selection set with less memory use.

The straightforward way to find the corresponding itemset is to do a top-down search in the subset lattice of the items. An example of such a subset lattice for five items is shown in Figure 1. The edges in this diagram indicate subset relations between the different itemsets.

To structure the search, we can organise the subset lattice as a prefix tree, which is shown in Figure 2. In this tree, those itemsets are combined in a node which have the same prefix with regard to a fixed order of the items. With this structure, the itemsets contained in a node of the tree can be easily constructed in the following way: Take all the items with which the edges leading to the node are labeled and add an item that succeeds after the last edge label on the path in the fixed order of the items. In this way, we only need one item to distinguish the itemsets after a particular node. Since many itemsets never happen, we only create the corresponding node when it occurs, saving a lot of memory. For example, the total number of all the possible diagnoses is 86. Theoretically, we need $2^{86}$ nodes to save all the possible combinations. But when we create the prefix tree for diagnoses, we only need around 0.1 million nodes.



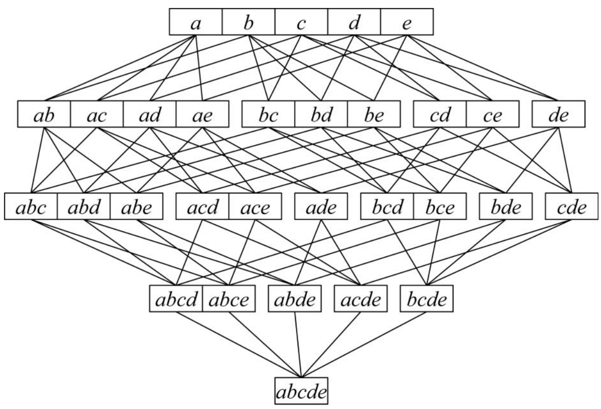


Figure 1. A subset lattice for five items.

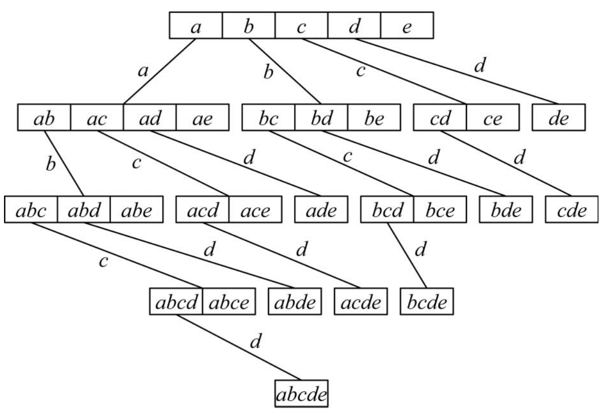


Figure 2. A prefix tree for five items.

## 4. Experiments

The dataset was extracted from a community hospital in the Midwest. Our experiments used 10,000 care plans as a training set and 5000 care plans as a testing set. We used the average ranking of selected items to do the evaluation. The best method has the minimal average ranking. Ideally, we hope it is equal to 1. It means we can always find a required item in the first place of the ranking list.

For the same care plan, different selection sequences may affect the average ranking. Suppose we are using the ranking list of support shown in Table 2 and we want to calculate the average ranking for the care plan containing only diagnoses 28 and 37. If we select 28 first, the ranking of 28 is 1. After 28 is chosen, it will be removed from the ranking list. The ranking of 37 will be bumped up to the 2nd position. In this sequence the average ranking is 1.5. If we select 37 first, the ranking of 37 is 3. After 37 is chosen, it will be removed from the ranking list either. However, the ranking of 28 is still 1. In this sequence, the average ranking is 2.

We use two different types of evaluation mechanisms, called *random selection* and *greedy selection*. Different selection methods generate different selection sequences. For random selection, we randomly select one item from the remaining items in the care plan and evaluate its ranking in the ordered list. For greedy selection, we always select the remaining care-plan item with the highest ranking in the list. Both of these can be seen as simulating human behaviour. When all required items are near the top of the list, human selection behaves like greedy selection. If all the required items are low in the list, people will not be patient enough to go through the list and would instead select the needed item in an alphabetic list. In this case human selection behaves more like random selection. Actual human selection is likely between the results of these two methods.

We compute the average ranking of selected items and report the results, averaged over five separate runs, in Table 3.

Given the poor performance of lift and entropy, we use the simple measure of support as the baseline for comparison, and both confidence and IV are better than support under both selection strategies. The comparison between confidence and information value is less obvious. Under the random selection strategy, the current selection does not affect future selections and confidence focuses only on minimising the ranking of the current selection. Intuitively, confidence is the best measurement under the random selection strategy. However, in the experiment the performance of information value is almost the same as that of confidence under random selection. In the greedy selection strategy, information value always does slightly better than confidence. The improvement is small but consistent. All differences are diluted by

Table 2. A support ranking list.

| Ranking | NANDA code | Selection description | Support value |
|---|---|---|---|
| 1 | 28 | Health maintenance | 0.87 |
| 2 | 52 | Knowledge deficit | 0.82 |
| 3 | 37 | Risk for infection | 0.55 |
| 4 | 12 | Pain acute | 0.53 |
| 5 | 39 | High risk for injury | 0.28 |
| 6 | 5 | Anxiety | 0.17 |

the existence of two disproportionately probable diagnoses that occur in nearly every care plan.

In order to examine the difference between confidence and information value in the greedy selection strategy, we repeat the experiment 100 times and compare the average ranking position of information value with that of confidence in the same experiment. In Figure 3, each point represents an experiment, the $x$-axis is the average ranking of information value, and the $y$-axis is the average ranking of confidence. Points above the line are experiments in which information value has a smaller average ranking than confidence. All the points in Figure 3 are above the line, i.e. information value outperforms confidence in each experiment. Moreover, information value has statistically significantly better performance ($p = 1.70313$E-60, using a pairwise $t$-test).

To examine what happens inside each method, we compute the average ranking of the selections in each iterative step of the selection process. In Figures 4 and 5, the $x$-axis represents the $i$-th step and the $y$-axis represents the average ranking value of choices made at that step. Under both greedy (Figure 4) and random (Figure 5)



Table 3. Average selection ranking.

| | 1 | 2 | 3 | 4 | 5 | Mean | Variance |
|---|---|---|---|---|---|---|---|
| Random selection | | | | | | | |
| Support | 5.396 | 5.338 | 5.439 | 5.434 | 5.341 | 5.390 | 0.049 |
| Confidence | 5.152 | 5.132 | 5.214 | 5.199 | 5.093 | 5.158 | 0.050 |
| Lift | 20.12 | 19.47 | 20.38 | 19.66 | 19.98 | 19.92 | 0.362 |
| Entropy | 38.54 | 38.27 | 39.07 | 38.83 | 38.48 | 38.64 | 0.314 |
| IV | 5.133 | 5.126 | 5.220 | 5.202 | 5.101 | 5.157 | 0.052 |
| Greedy selection | | | | | | | |
| Support | 4.320 | 4.292 | 4.397 | 4.382 | 4.287 | 4.336 | 0.051 |
| Confidence | 3.905 | 3.909 | 3.990 | 3.998 | 3.897 | 3.940 | 0.050 |
| Lift | 15.81 | 15.63 | 16.18 | 15.76 | 15.78 | 15.83 | 0.206 |
| Entropy | 31.66 | 32.60 | 32.58 | 32.49 | 31.95 | 32.26 | 0.426 |
| IV | 3.895 | 3.898 | 3.986 | 3.988 | 3.880 | 3.929 | 0.053 |

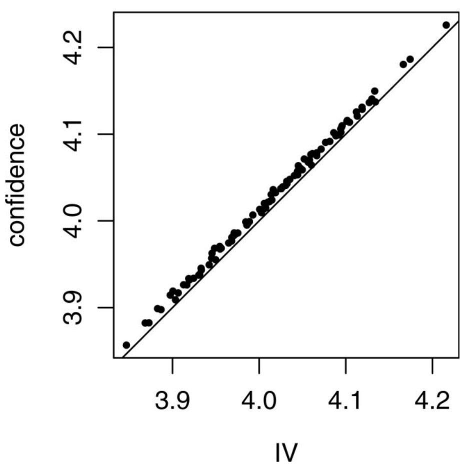


Figure 3. Information value vs. confidence.

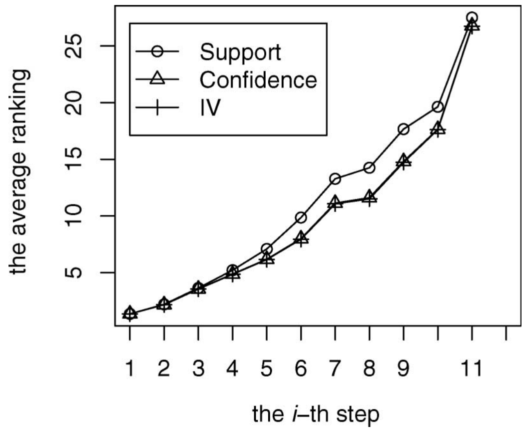


Figure 4. The average *i*-th step result of greedy selection.

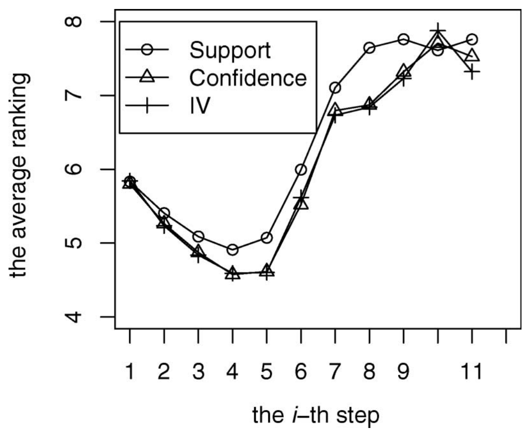


Figure 5. The average *i*-th step result of random selection.



selection, both confidence and information value are consistently better than support. Since the performance difference between confidence and IV is difficult to see, we calculated the difference between them in each step, as shown in Figures 6 and 7. Under greedy selection, the performance of information value is constantly better than that of confidence, increasing through the 8th selection. After that, the improvement decreases but is still positive. However, no such pattern is evident under random selection, and overall there is no difference between the two values. Figures 6 and 7 support the conclusion that the performance of information value is almost the same as that of confidence in the random selection strategy and consistently better than confidence under greedy selection.

Right after the above experiments, we also conducted the similar experiments by using the intervention data. For diagnoses, the total number of possible items is 87, while for interventions the total number is 250. We get the similar results. Average ranking of support is 18.67; average ranking of confidence is 14.02; and average ranking of IV is 13.99. We also examined the trade-off between confidence (immediate probability) and entropy (future probability) in the information value measurement,

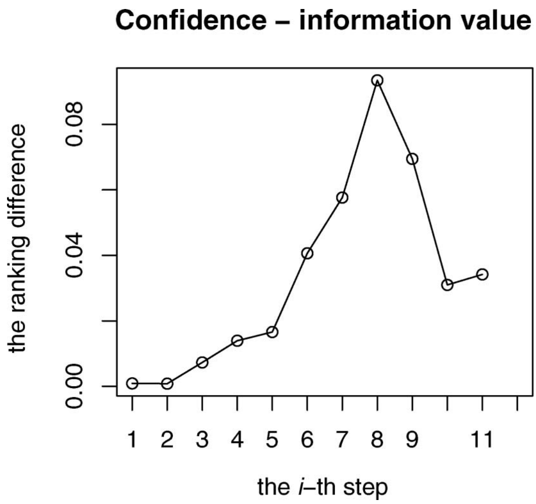


Figure 6. The difference in the *i*-th step result of greedy selection.

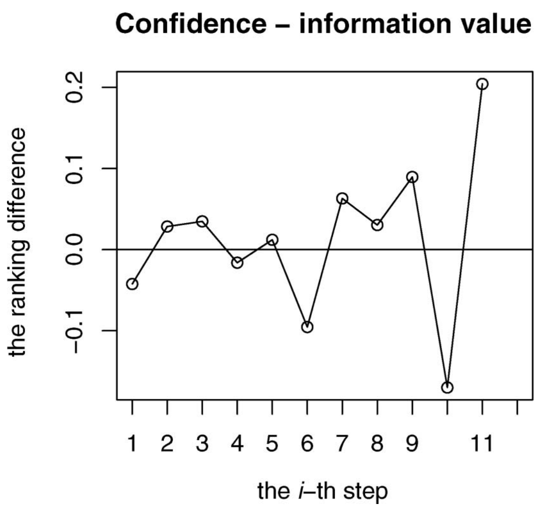


Figure 7. The difference in the *i*-th step result of random selection.

and adjusted it to perform better on specific problems. In order to adjust the trade-off between confidence and entropy, we adjusted our ranking measure to the following formula: $\lambda \times$ *confidence* $+ (1 - \lambda) \times (1 -$ entropy$)$. However, it turns out that no matter how we adjust the value of $\lambda$ , the final result does not exceed the original multiplicative formula. In the future, we will try to adjust the weight in the following formula: $confidence^{\lambda} \times (1 - \text{entropy})^{(1-\lambda)}$ to find better trade-off.

## 5. Conclusion and future work

We have described a new recommendation technique based on several measurements. In addition to traditional measurements support and confidence, we also test the effectiveness of a novel measurement – information value – which balances the gain between current and future selections. Its performance surpasses that of confidence, and it is still computable in real time. Such a system is a complement to expert systems and traditional practice guidelines, and can be very useful for nursing education and clinical quality control. It has the capability to connect systems, users, nursing care process and information in a way that allows nursing care to become more connected and responsive (Erol *et al.* 2010). Such a system also pays attention on capturing information of users, tasks and services which can be used for recommendation (Wang *et al.* 2010).

The effectiveness difference between expert systems and such a recommender is also interesting. Rules from experts' knowledge could be more accurate but they are not easily updated and specialised for different hospitals. Can we combine these two kinds of systems to achieve better results?

Another promising direction is to incorporate contextual information into the recommendation process and make recommendations based on multiple dimensions, patient profiles, and other information (Adomavicius *et al.* 2005). One unexplained point in the current experiment is how we were able to get the ranking gain even in the first step. Originally, we expected to sacrifice some of the current gains for the future gain. The final result contradicted our prediction. A reasonable explanation for this result is that the information value formula indirectly increases the diversity on the top of the ranking list. When we have two highly correlated items to select, only one of them is needed on the top of the ranking list once the other item is on the top of the subsequent ranking list. This method can improve the ranking position of other items in the current ranking list without jeopardising the ranking of the first two items. In the future, we hope to increase the diversity on the top of the ranking list in order to decrease the average ranking. Finally, as we keep adding more items into our current care plan, the sample space containing previous selected items shrinks exponentially. When the sample space is less than 50, it is statistically less reliable for us to calculate all the proposed measurements; however, a care plan could be a combination of several patient phenomena. Given a previous set of selected items, we hope to segment this given set into several smaller sets. Each segmented small set is related to separated patient phenomenon. We can recommend based on each segmented set. By doing this, we might relieve the exponentially shrunk sample space problem.



## References

Adomavicius, G., *et al.*, 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information System*, 23 (1), 103–145.

Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734–749.

Balabanović, M. and Shoham, Y., 1997. Fab: content-based, collaborative recommendation. *Communication of ACM*, 40 (3), 66–72.

Bellika, J. and Hartvigsen, G., 2005. The oncological nurse assistant: a web-based intelligent oncological nurse advisor. *International Journal of Medical Informatics*, 74, 587–595.

Billsus, D., *et al.*, 2002. Adaptive interfaces for ubiquitous web access. *Communication of the ACM*, 45 (5), 34–38.

Borgelt, C., 2003. Efficient implementations of a priori and eclat. *In: Proceedings of IEEE ICDM the workshop on frequent item set mining implementations*, Melbourne, FL. 90–96.

Dochterman, J.M. and Bulechek, G.M., 2003. *Nursing interventions classification*. St. Louis, MO: Mosby.

Erol, O., Sauser, B., and Mansouri, M., 2010. A framework for investigation into extended enterprise resilience. *Enterprise Information Systems*, 4 (2), 111–136.

Han, J., 2005. *Data mining: concepts and techniques*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Hardiker, N., *et al.*, 2002. Formal nursing terminology systems: a means to an end. *Journal of Biomedical Informatics*, 35, 298–305.

Herlocker, J.L., *et al.*, 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22, 5–53.

Hill, W., *et al.*, 1995. Recommending and evaluating choices in a virtual community of use. *In: CHI '95*: *Proceedings of the SIGCHI conference on human factors in computing systems*, Denver, Colorado, United States. New York, NY: ACM Press/Addison-Wesley Publishing Co., 194–201.

Iglesias, A., Martnez, P., and Fernndez, O., 2003. *Err is human: building a safer health system*. Vol. 5. Athena. National Academy Press; 2000, 223–240.

Kakousis, K., Paspallis, N., and Papadopoulos, G., 2010. A survey of software adaptation in mobile and ubiquitous computing. *Journal of Enterprise Information Systems*, 4 (4), 355–389.

Keenan, G.M., Yakel, E., and Marriott, D., 2006. HANDS: a revitalized technology supported care planning method to improve nursing handoffs. *In: 9th international congress on nursing informatics, Seoul, South Korea*, The Netherlands: IOS Press, 580–584.

Koren, Y., 2008. Tutorial on recent progress in collaborative filtering. *In: RecSys '08*: *Proceedings of the 2008 ACM conference on recommender systems*, Lausanne, Switzerland. New York, NY: ACM, 333–334.

Li, L., *et al.*, 2008. Creation of environmental health information system for public health service: a pilot study. *Information Systems Frontiers*, 10, 531–542.

Li, L.X. and Xu, L.D., 1991. An integrated information system for the intervention and prevention of AIDS. *International Journal of Bio-Medical Computing*, 29 (3–4), 191–206.

Linden, G., Smith, B., and York, J., 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7 (1), 76–80.

Luo, J., *et al.*, 2007. Flood decision support system on agent grid: method and implementation. *Enterprise Information System*, 1, 49–68.

MacKinnon, W., and Wasserman, M., 2009. Integrated electronic medical record systems: critical success factors for implementation. *In: System sciences, 2009*. HICSS '09. 42nd Hawaii international conference. Hawaii: IEEE CS Press, 1–10.

McNee, S.M., Riedl, J., and Konstan, J.A., 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. *In: CHI '06: CHI '06 extended abstracts on human factors in computing systems*, Montréal, Québec, Canada. New York, NY: ACM, 1097–1101.

Miller, B.N., *et al.*, 2003. MovieLens unplugged: experiences with an occasionally connected recommender system. *In: IUI '03*: *Proceedings of the 8th international conference on intelligent user interfaces*, Miami, Florida, USA. New York, NY: ACM, 263–266.

Moorhead, S., *et al.*, 2005. *Nursing outcomes classification*. St. Louis, MO: Mosby.

Nanda, 2005. *Nanda nursing diagnoses: definitions and classification*. Philadelphia: PA, Nusercom Inc.

Puustjarvi, J. and Puustjarvi, L., 2010. Developing interoperable semantic e-health tools for social networks. *In: Proceedings 6th international conference on web information systems and technologies*, Valencia, Spain. Setubal, Portugal: INSTICC Press, 305–312.

Resnick, P., *et al.*, 1994. GroupLens: an open architecture for collaborative filtering of netnews. *In: Proceedings of ACM 1994 conference on computer supported cooperative work*. Chapel Hill, NC: ACM Press, 175–186.

Ryan, S.A., 1985. An expert system for nursing practice. *Journal of Medical Systems*, 9 (1–2), 29–41.

Shardan, U. and Maes, P., 1995. Social information filtering: algorithms for automating 'word of mouth'. *In:* Irvin R. Katz, R. Mack, L. Marks, M.B. Rosson and J. Nielsen, eds. *Proceedings of the SIGCHI conference on human factors in computing systems (CHI '95)*. New York, NY: ACM Press/Addison-Wesley Publishing Co., 210–217.

Wang, J.W., Gao, F., and Ip, W.H., 2010. Measurement of resilience and its application to enterprise information systems. *Enterprise Information System*, 4, 215–223.

Xu, L.D., 1994. A decision support system for AIDS intervention and prevention. *International Journal of Bio-Medical Computing*, 36 (4), 281–291.

Yoo, S.K., Choe, J., and Kim, D.Y., 2008. Agent based architecture for secure access from multiple hospitals. *In: Proceedings of the Seventh IEEE/ACIS international conference on computer and information science (ICIS 2008)*. Washington, DC, USA: IEEE Computer Society, 255–258.