Contrast Pattern Based Collaborative Behavior Recommendation for Life Improvement

Yan Chen[1(✉)], Margot Lisa-Jing Yann[1], Heidar Davoudi[1], Joy Choi[1], Aijun An[1], and Zhen Mei[2]

[1] Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
{ychen,lisayan,davoudi,aan}@cse.yorku.ca, ysjoychoi@gmail.com
[2] Manifold Data Mining Inc., Toronto, Canada
zhen@manifolddatamining.com

Abstract. Positive attitudes and happiness have major impacts on human health and in particular recovery from illness. While contributing factors leading human beings to positive emotional states are studied in psychology, the effects of these factors vary and change from one person to another. We propose a behaviour recommendation system that recommends the most effective behaviours leading users with a negative mental state (i.e. unhappiness) to a positive emotional state (i.e., happiness). By leveraging the contrast pattern mining framework, we extract the common contrasting behaviours between happy and unhappy users. These contrast patterns are aligned with user behaviours and habits. We find the *personalized* behaviour recommendation for those with negative emotional states by placing the problem into the nearest neighborhood collaborative filtering framework. A real dataset of people with heart disease or diabetes is used in our recommendation system. The experiments conducted show that the proposed method can be effective in the health-care domain.

1 Introduction

The pursuit of happiness can be characterized as a psychological factor and a life goal for human beings. The emerging field of sentiment analysis and opinion mining provides a means of computational analysis of emotion, affect, subjective experience and perception. These factors have a direct effect on human behaviours and attitudes. However, how the factors affect people psychologically is not apparent.

Personalized health-care can improve the patients' health experience and prognosis; early intervention can significantly reduce the health-care cost caused by related and predictable emergent conditions. Our goal is to provide recommendations of the most effective behaviours leading to positive psychological attitudes for high-risk patients with chronic diseases, in order to reduce health-care cost by reducing, e.g., the incidence of acute treatment related to mismanagement of disease conditions.

J. Kim et al. (Eds.): PAKDD 2017, Part II, LNAI 10235, pp. 106–118, 2017.
DOI: 10.1007/978-3-319-57529-2_9

The targeted behaviour recommender system faces several challenges. First, traditional recommender techniques work based on the notion of implicit/explicit rating for a set of items. These ratings are not available for a behaviour recommendation system, but rather each user is characterized as a set of attribute-values. Second, latent factor models can capture underlying reasons behind the user behaviour/preference even though it could be quite difficult for them to recommend behaviours that cannot be characterized by the latent factors. Third, recommendation systems are usually evaluated based on the standard measures such as precision@N and Mean Square Error (MSE). However, due to different problem settings and lack of the ratings we need an intuitive evaluation protocol for this type of recommendation systems.

Due to these challenges for personalized health-care, we need a novel methodology to provide effective recommendation, and to give a personalized evaluation system for improving health. Thus, our focus is on analyzing the high-risk patients with chronic disease related lifestyle and social conditions, as well as identifying the difference that exists between positive and negative attitudes. We use contrast pattern mining on a rich dataset that includes a population of patients with heart disease or diabetes, to identify group behavioural factors that reflect an individual's emotional state of unhappiness or happiness. With the contrast patterns, we generate recommendations based on the existing differences among the population. This information of contrast patterns describing the difference is used to build a behavioural recommendation system that provides recommendations for individuals with attitudes to make certain behaviour changes. In order to find the most relevant recommendations, the k-nearest neighbours (k-NN) algorithm is applied to identify the most effective behaviours for the user from the contrast patterns found.

The major goal of our proposed recommendation system is to discover and recommend the behaviours to improve the quality of life of users. As such, the problem of behaviour recommendation can be defined so as to provide users with recommendations based on the differences extracted between groups of people, in order to improve their lifestyle and life satisfaction.

The contributions of this paper are as follows:

- We define and formulate the problem of behaviour recommendation and design an effective solution for it.
- We apply contrast pattern mining to identify the transitional patterns as effective recommendations (i.e., behaviours), to suggest users to become members of a class of interest (i.e., happy people).
- A simple intuitive protocol based on standard evaluation methods is designed to assess the effectiveness of these types of recommenders.
- We conduct the experimental evaluation and show the effectiveness of proposed model in a health-care domain.

2 Related Work

Recommendation systems have been widely utilized in different domains to meet user interests and boost user satisfaction. For example, recently Abel proposed a recommendation system to help people find a job [1], or Backstrom and Leskovec [3] suggested a recommender system to find friends from social networks (i.e., Facebook). However, to date, most of recommendation systems have been applied in e-commerce and news domains [3,11,15]. Approaches for recommendation systems are usually divided in three broad categories: *collaborative filtering* [15], *content-based* [12] and *hybrid* [4] approaches. In collaborative filtering, we recommend items in which people with similar tastes and preferences are interested. Furthermore, the collaborative filtering techniques can be categorized into two general classes of *neighborhood* and *model-based* methods. In neighborhood-based (i.e., memory-based/heuristic-based) methods, user ratings for items stored in the system are directly used to generate the list of recommendations or predict the ratings for new items. Two major approaches in this framework are *user-based* collaborative filtering [15] whereas interest of a user for an item is estimated based on the rating for this item by other users (i.e., neighbors), and *item-based* approaches [11] which predict the rating of a user for an item based on the rating of the user for similar items. In contrast to neighborhood-based methods, model-based approaches exploit the users ratings to learn a predictive model. Bayesian Clustering [4], Latent Semantic Analysis [8], Latent Dirichlet Allocation [17], and Maximum Entropy are instances of this category. On the other hand, content-based recommender systems [9] recommend the items that are similar to the ones that she/he was interested in the past, and hybrid approaches refer to the class of algorithms that combine collaborative and content-based schemes to achieve better performance.

Another related area is contrast pattern mining. Contrast patterns are those that are significantly different among different classes, times, locations or/and other dimensions of interest. They have been utilized in different tasks and applications such as building the accurate and robust classifiers [14], detecting malware [16], or diagnosing disease [10]. The contrast patterns reflecting different frequencies in two datasets sometimes are refereed as *diverging patterns* [2], or *emerging patterns* [13]. For example, An et al. [2] consider a pattern as the diverging if its respective supports in two datasets and its diverging ratio (defined based on the distance between the four-dimensional vectors representing pattens) is more than certain thresholds. Ramamohanarao and Bailey [13] suggested different types of emerging patterns such as jumping emerging patterns (which exist in one dataset and are absent in another one), constrained emerging patterns (whose supports are more and less than specific thresholds in the first and second dataset accordingly). They argued while jumping emerging patterns can represent the sharp contrast between two datasets, they are susceptible to noise, so in many cases constrained emerging patterns would be the better choices. Webb et al. [18] proposed that contrast pattern mining can be seen as a special case of the general rule learning task where contrast patterns and groups for which they are characteristic are the antecedents and consequents of the rules respectively. This formulation allows any standard rule discovery algorithm to be adapted for the contrast pattern mining problem.

All aforementioned recommendation methods work based on the notion of implicit/explicit ratings of users for items. However, for the behavior recommendation problem such ratings are not available. Moreover, to best of our knowledge, there is no work on using contrast pattern mining for recommendation purpose. In contrast, in our problem setting, users are characterized based on set of attribute-values and belong to one of two disjoint classes (i.e., happy and sad people). The goal is to recommend a set of most effective transitional patterns (i.e., behaviors) which make a user likely to become a member of the class of interest (i.e., happy people).

3 Methodology

In this section, the dataset used in our system is first described. Then we present the overall framework of our recommendation system. The detailed steps for generating recommendations will be discussed in different subsections.

3.1 Dataset Description

The dataset comes from 2011/2012 Canadian Community Health Care Survey Data, which includes 16836 patients with diabetes and heart disease. The attributes (and their respective values) of patients in the dataset are captured with more than 100 survey questions. These questions are classified into seven categories, namely, geo-demographics, lifestyles, adherence, health-care experience, mental health, social connections and supports, and quality of life.

In the original dataset, the data is first discretized and transformed into transaction dataset with itemsets, where each item is an attribute-value pair. Furthermore, based on the characteristics of the attributes, we categorize the attributes into three different types:

– **static** attributes: cannot be changed, e.g. gender, age, or suffering from heart disease.
– **mutable** attributes: can be changed, e.g. alcohol use, volunteering activities, characteristic and habitual behaviors that signify mood or attitude.
– **swing** attributes: can or cannot be changed depending on willingness, ability to undertake cognitive behavioral change or other factors.

Not all of the attributes in this dataset have significant affect on the "happiness" of people. The attributes are filtered to remove the insignificant ones. Only 30 of the attributes are left in the dataset. Weka [7] is used here for this purpose, with the built-in "AttributeSelection" filter.

Table 1 shows some examples of attributes and values in this dataset.

3.2 Overall Framework

Figure 1 shows the overall framework of our recommendation system. Generally, our system includes the following steps for the process of generating recommendations:

Table 1. Attributes and values examples

Category	Attribute	Value
Life style	Daily consumption (fruits/vegetables)	5-10 Times/day
Life style	Smoke	Daily
Mental health	Satisfaction with life in General	Satisfied
Mental health	Perceived life stress	Not at all
Quality of life	Pain	No pain
Geo-demographics	Health Region	City of Toronto
Healthcare experience	No. of consultations with medical doctor	Not at all

a. Generate contrast patterns;
b. For each individual in dataset, find all the matching contrast patterns;
c. Find k-nearest neighbors for the current user;
d. Provide recommendation to the current user from its neighbors' matching contrast patterns;

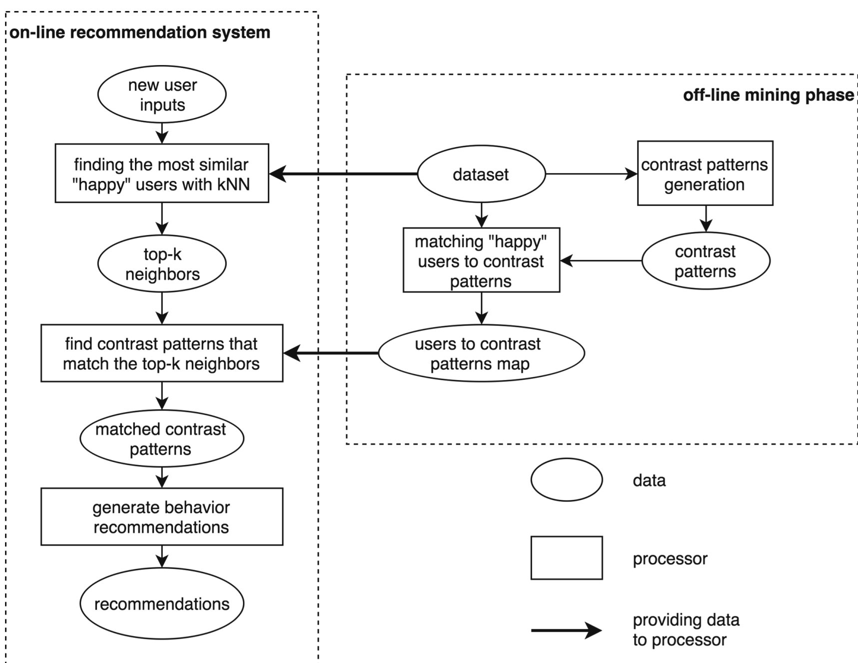


Fig. 1. Recommendation system flow chart

We have two main stages in our recommendation. The off-line stage includes contrast pattern mining and the process of matching users to contrast patterns. In order to distinguish the underlying differences between happy and unhappy, we use contrast pattern mining on the dataset to identify groups of behavior factors that may change people's feeling from unhappy to happy or vice versa. The contrast patterns extracted from the dataset can be applied to all the users in the dataset and need to be *personalized* for each individual user. We refer to these patterns as *global* contrast patterns in this paper. With such information, we utilize neighborhood-based collaborative filtering framework to customize recommendations for each user to adopt contrasting groups of behaviors. Last, in our on-line recommendation system, upon completion of a questionnaire by a user, user similarity assessment is performed using the k-nearest neighbors algorithm to identify people that are similar to the user. Subsequently, *global* contrast patterns (i.e., the contrast patterns found over the whole dataset) that match the identified similar people are used to generate personalized behavior recommendations that will have positive impacts on the user.

Pseudo-code of our behavior recommendation system is provided in Fig. 2.

```
Input: S+, S− (Off-line Phase)        ▷ S+, S−: Set of happy and unhappy people
       u (On-line Phase)              ▷ u: a user asking for recommendation
Output: R                             ▷ R: Set of behavior recommendations

function
Off-line Phase:
    C_P ← Generate Contrast Pattern(S+, S−)
    U_CP ← Map User to Contrast Pattern(U, C_P)
On-line Phase:
    R ← ∅
    U_NN ← Find Nearest Neighbours(u)
    for all v ∈ U_NN do
        X_cp ← Retrieve(U_CP, v)      ▷ Contrast patterns to which user u maps
        R ← R ∪ X_cp
    end for
    Resolve Conflicting Attributes    ▷ as described in section 3.6
    Remove Static Attributes
    return R
end function
```

Fig. 2. Pseudo-code of our behavior recommendation system.

3.3 Contrast Patterns Generation

Before we describe the process of generating contrast patterns, some definitions will be given first.

Let $I^* = \{I_1, I_2, ..., I_m\}$ be a set of items. An itemset X is a set of items $\{I_{e_1}, I_{e_2}, ..., I_{e_Z}\}$, where Z is the length of X, denoted by $|X|$. A dataset D is a list of transactions $\{T_1, T_2, ..., T_n\}$, where each transaction $T_d \in D$ is an itemset.

Definition 1 (**Support of itemset X in dataset D**)**.** The support of itemset X in D is defined as the fraction of itemsets in D, which contain itemset X.

Definition 2 (**Contrast ratio of itemset X in datasets D_1 and D_2**)**.** The contrast ratio of itemset X in D_1 and D_2 is defined as $\frac{support\ of\ X\ in\ D_2}{support\ of\ X\ in\ D_1}$.

Please note that the order of the datasets in the Definition 2 affects the contrast ratio. Inverting the order will also invert the ratio value.

In our recommendation system, we use the definition of contrast pattern as well as the mining algorithm for contrast pattern mining from Fan and Ramamohanarao [5].

Definition 3 (**Contrast pattern in datasets $D_1$and D_2**)**.** An itemset X is a contrast pattern in datasets D_1 and D_2, if and only if

1. *Contrast Ratio of* $X \geq threshold_1$ (*denoted as* θ *in Sect.* 4),
2. *Contrast Ratio of* $X \geq$ *Contrast Ratio of* $Y \quad \forall\, Y \subseteq X$,
3. *Support of* X *in* $D_1 \geq threshold_2$ & *Support of* X *in* $D_2 \geq threshold_2$,
4. $\chi^2 \geq threshold_3$.

In Definition 3, condition 1 filters out patterns with low ratios, which corresponds to non-effective patterns. In condition 2, we ensure that every item in the contrast pattern contributes to higher contrast ratios. If a pattern is already a contrast pattern and adding a new item into the pattern decreases the contrast ratio, the new itemset should not be a contrast pattern even if the new contrast ratio is still above the threshold. In addition, we also want to find contrast patterns representing relative broader popularity, instead of just a small group of people. That's why we have condition 3 to remove patterns with low supports. The last condition evaluates the correlation of internal items using chi-square value measure, which ensures that the items in the contrast patterns are actually strongly correlated. The contrast pattern mining algorithm [5] employs a tree structure and identifies contrast patterns efficiently both in terms of memory and time.

Contrast patterns found among the different classes in the population are the originating sources for our recommendation system. As these patterns show the most significant behavioral difference leading to class changes. Specifically, if one itemset is a contrast pattern, it means the occurrences of this itemset for the 'happy' class and 'unhappy' class are markedly different. In other words, if someone conforms with this contrast pattern, this person will be much more likely to belong to one class than the other. For example, if percentages of people who 'smoke' and 'consume little fruit in their diet' appears in the positive class and negative class are 1% and 12% respectively, we can conclude that, people who 'smoke' and 'consume little fruit' are 12 times more possible to express a negative attitude. This "smoke and consume little fruit" is a contrast pattern and its contrast ratio is 12. Given these contrast patterns, each contrast pattern is converted into a set of recommendations. For example, given the the contrast pattern of "smoke and consume little fruit", one individual with negative 'unhappy' class label is given advice to avoid "smoke and consume little fruit" as an alternate lifestyle choice. In general, numerous recommendations are generated based on the contrast patterns found among the population. Specifically, over 4,000 contrast itemsets are generated from our dataset, most of which have more than two items in each pattern.

3.4 Matching "Happy" Users to Contrast Patterns

Given the contrast patterns for the positive and negative class, we need to map users to the contrast patterns. As such, each user in the training dataset is compared to all contrast patterns. The matching process is fairly simple. A user is matched to a contrast pattern if she/he has all attribute-values of the contrast pattern. For example, an individual maps to the contrast pattern of 'smoke and consume little fruit' if he/she 'smokes' and 'consumes little fruit'.

3.5 Finding the Most Similar "Happy" Users with kNN

The *global* contrast patterns show the significant differences between two population of users even though they are not personalized for individual users and, consequently cannot be used directly for the recommendation purpose. As such, we adopt the neighborhood-based collaborative filtering approach to find the set of personalized recommendation candidates. In particular, with a new user's inputs, k 'happy' users who are most similar to her/him are identified. We use Pearson's correlation coefficient to calculate the similarities between two individuals as it can handle missing values and grade-inflation well [6] (there are a lot of missing values in the dataset we use since users usually do not answer all questions).

3.6 Behavioral Recommendations Generation

Given the k nearest neighbors of a new user, the set of contrast patterns to which each neighbor user maps is obtained using the results from Sect. 3.4. Next, the union of all contrast patterns of the k neighbors is considered as the initial set of personalized recommendation candidates. It is possible that we have some conflicting attribute-values (e.g., job type attributes is set to both part time and full time), in that case, we keep the the attribute-value with the higher contrast ratio rate. In case that the ratio rates are the same, the conflicting attribute-values are chosen randomly. Furthermore, we remove the static attributes (as they are not appropriate for behavior recommendation) and the existing attributes (those that the user already has) from the set of recommendation candidates and generate the final set of recommendations. The rationale is that while there are some behaviors which are prevalent among happy users, certain behaviors are common among specific group of happy users. As such, The more similar user U_1 is to happy user U_2, the more likely that user U_2 behaviors can be applied to user U_1 to achieve happiness. For example, if user U_1 is similar to user U_2 and U_3, and user U_2 and U_3 are mapped to set of contrast patterns S_2 and S_3 respectively, the initial set of recommendation candidates is $S_1 \cup S_2$.

It is worth to mention that if an attribute is a swing attribute, we provide an extra question in our recommendation system asking whether the user is able/willing to make changes on this attribute. The attribute is then categorized into static or mutable according to users' answers.

4 Experimental Results and Analysis

In this section, our recommendation system will be evaluated. The evaluation focuses mainly on the effectiveness of our off-line training approach. The implementation details of the on-line recommendation system are also mentioned at the end of this section.

4.1 Evaluation Design and Protocol

Recommendation systems traditionally are evaluated based on the classification performance measurement (e.g., Precision @ 10), rank-based performance measurement, or rate-based performance measurement (e.g., MSE), depending on the targeted task (e.g., predicting the top recommendation item or rates). However, for the behavior recommendation task such ranks and the ground truth are not available. In fact, we do not know whether the recommendations of particular behaviors will make users happy or not in the real life. As such, one major challenge in the proposed behavior recommendation system is how to evaluate it.

In order to address the evaluation problem, and measure the effectiveness of our recommendation system, we develop a classification system to calculate the possibility of being 'happy' for a user before and after applying the recommendations. If the possibility of being happy increases after applying the recommendations, it means the recommendations are effective. We compare different classification techniques and choose an ensemble method as user class (i.e., happy or unhappy) predictor. The classification algorithm is ensemble of AdaBoost, Random Forest, J48, Bayesian Network, and Logistic Regression. The final results are based on the voting method (the same weight for all the algorithms are used). For this part we use the standard implementation of these algorithms Weka [7] with their respective default parameters. Using 10-fold cross validation, the overall accuracy of the ensemble method reaches 71.8%.

4.2 Performance Evaluation

To effectively evaluate our recommendation system, we designed two other approaches as baseline, called *RANDKN* and *TOPCP*, for comparison purposes. The *RANDKN* approach does not use k-NN to find the most similar users to the new user. Instead, it tries to find the same number of contrast patterns as in k-NN randomly. For example, for a new user, in our recommendation system, if 20 contrast patterns are generated and applied to the user, the system also choose 20 contrast patterns in *RANDKN*, but randomly. *TOPCP* uses the same strategy as *RANDKN*, but instead of choosing contrast patterns randomly, *TOPCP* ranks the contrast patterns by their ratios and chooses the same number of contrast patterns as in our recommendation system with highest ratios.

In the evaluation, we first use leave-one-out to generate recommendations for each user in the dataset. Then the recommendations are applied to the user.

Table 2. Effectiveness of recommendation systems (A: percentage of people having probability of happy improved; B: percentage of unhappy instances having probability of happy improved).

Algorithm	A	B
Our method	66.03%	95.82%
RANDKN	60.82%	91.94%
TOPCP	58.95%	89.69%

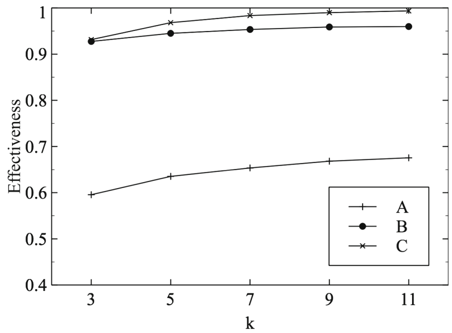
Thereafter, a new dataset containing all the itemsets after applying the recommendations is generated. The ensemble method described previously is used to evaluate the classification of each user as type '1' (happy) or type '2' (unhappy) on the new dataset. The possibility of being classified as type '1' (happy) for each date sample is computed. The results are shown in Table 2.

As shown in Table 2, we can see that 95.82% of the unhappy users may become happier after applying the recommended changes using our method. Comparing the *RANDKN* (random selected k neighbors) and *TOPCP* (selected 20 highest-ratio neighbors), our method performs better than both of them. The fact that *TOPCP* performs worse than *RANDKN* is because the contrast patterns with the highest ratios can have significant amount of overlaps, which decreases the number of item choices in the recommendations. This further justifies that our approach of using k-NN to identify most similar neighbors in order to further obtain related contrast pattern items.

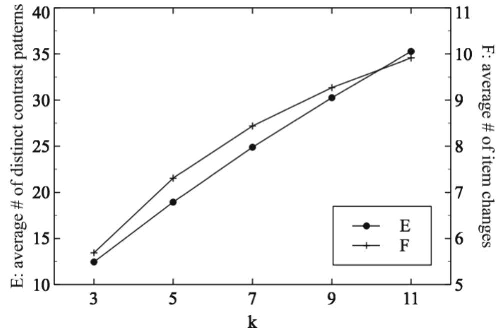
In the experiments, we choose $k = 7$ and $\theta = 4.0$. Note that we do not set the value of θ too low, since lower θ values lead to less effective patterns. θ can also not be set too high; otherwise, not enough patterns are given as recommendation to users. Thus, $\theta = 4.0$ is chosen so that the patterns are enough for recommendations and effective in the same time. The k value for k-NN method is also carefully selected. If a higher k is chosen, the user is recommended with more patterns. Too many patterns are not practical for users to take on all the recommendations. But too few also does not provide enough information for the user to obtain possible recommendations to change to be happier. We run experiments on using different k values, and compared the number of distinct contrast patterns, the number of item changes, percentage of people having probability of happy improved and percentage of unhappy instances having probability of happy improved. The results are in Fig. 3.

4.3 Implementation

The initial version of implemented system uses Django as web framework and MySQL as data storage layer. However, it may take up to several minutes to generate recommendations for a single user on a server with a Intel(R) Xeon(R) CPU E5-2620 v3 CPU and 64 GB of RAM. The reason is that the system needs to scan the complete dataset for every user to find the k nearest neighbors. Also, the



(a) Effectiveness
A: percentage of people having probability of happy improved;
B: percentage of unhappy instances having probability of happy improved;
C: percentage of unhappy instances having probability of happy over 50%



(b) Recommendation size
E: average # of distinct contrast patterns;
F: average # of item changes

Fig. 3. Performance of recommendation system with different k values

program design only uses a single thread, which means that only a small fraction of the computation power of the CPU is used. To solve the performance issue mentioned above, we refactor the implementation to use all cores of the CPU on the server. Furthermore, we design to enable our recommendation system to be able to scale across different machines in order to allow large numbers of users to use this website in the same time. In the refactored website, we use Play framework and Akka to distribute the computation across a cluster of servers. Redis has been used for the data storage to allow us to retrieve all the data from memory instead of disk to reduce the processing time. After applying all changes, the system takes approximately 1–5 s for each user to obtain its recommendations online.

5 Conclusions

The impact of positive attitudes is an acknowledged factor in people's health. However, the contributing factors to happiness depends on characteristics which are unique and subjective. In this paper, we proposed a personalized behaviour recommendation that recommends the most effective behaviours for changing users emotional state from negative (i.e., unhappy) to positive (i.e., happy). We showed that contrast pattern mining served effectively as the transitional patterns from the negative to positive class. The contrast pattern mining framework is adopted and combined with the collaborative filtering to produce the personalized behaviour recommendation. The experiments on an actual dataset showed that our proposed method performed well and was effective in the health-care domain.

Acknowledgement. This work is funded by the Big Data Research, Analytics, and Information Network (BRAIN) Alliance (established by the Ontario Research Fund - Research Excellence Program), Manifold Data Mining Inc., and Natural Sciences and Engineering Research Council of Canada (NSERC). We would like to thank Manifold for providing the dataset used in this research. In particular, we thank Ted Hains of Manifold and Jianhong Wu of York University for their insights and collaboration in our joint project.

References

1. Abel, F.: We know where you should work next summer: job recommendations. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 230–230. ACM (2015)
2. An, A., Wan, Q., Zhao, J., Huang, X.: Diverging patterns: discovering significant frequency change dissimilarities in large databases. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1473–1476. ACM (2009)
3. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 635–644. ACM (2011)
4. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. **12**(4), 331–370 (2002)
5. Fan, H., Ramamohanarao, K.: Efficiently mining interesting emerging patterns. In: Dong, G., Tang, C., Wang, W. (eds.) WAIM 2003. LNCS, vol. 2762, pp. 189–201. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45160-0_19
6. Garren, S.T.: Maximum likelihood estimation of the correlation coefficient in a bivariate normal model with missing data. Stat. Probab. Lett. **38**(3), 281–288 (1998)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
8. Karacapilidis, N., Hatzieleftheriou, L.: A hybrid framework for similarity-based recommendations. Int. J. Bus. Intell. Data Min. **1**(1), 107–121 (2005)
9. Kim, W., Kerschberg, L., Scime, A.: Learning for automatic personalization in a semantic taxonomy-based meta-search agent. Electron. Commer. Res. Appl. **1**(2), 150–173 (2002)
10. Li, J., Yang, Q.: Strong compound-risk factors: efficient discovery through emerging patterns and contrast sets. IEEE Trans. Inf. Technol. Biomed. **11**(5), 544–552 (2007)
11. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
12. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). doi:10.1007/978-3-540-72079-9_10
13. Ramamohanarao, K., Bailey, J.: Emerging patterns: mining and applications. In: Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004, pp. 409–414. IEEE (2004)
14. Ramamohanarao, K., Bailey, J., Fan, H.: Efficient mining of contrast patterns and their applications to classification. In: 3rd International Conference on Intelligent Sensing and Information Processing, pp. 39–47. IEEE (2005)

15. Ricci, F., Rokach, L., Shapira, B.: Introduction to Recommender Systems Handbook. Springer, Heidelberg (2011)
16. Sun, X., Huang, Q., Zhu, Y., Guo, N.: Mining distinguishing patterns based on malware traces. In: 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 2, pp. 677–681. IEEE (2010)
17. Trewin, S.: Knowledge-based recommender systems. Encycl. Libr. Inf. Sci. **69**(Suppl. 32), 180 (2000)
18. Webb, G.I., Butler, S., Newlands, D.: On detecting differences between groups. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 256–265. ACM (2003)