Maria Rafalak
Janusz Granat
Andrzej P. Wierzbicki

APPLICATION OF MULTI-CRITERIA ANALYSIS BASED ON INDIVIDUAL PSYCHOLOGICAL PROFILE FOR RECOMMENDER SYSTEMS

Abstract *This paper presents a novel approach for user classification exploiting multi-criteria analysis. This method is based on measuring the distance between an observation and its respective Pareto front. The obtained results show that the combination of the standard KNN classification and the distance from Pareto fronts gives satisfactory classification accuracy – higher than the accuracy obtained for each of these methods applied separately. Conclusions from this study may be applied in recommender systems where the proposed method can be implemented as the part of the collaborative filtering algorithm.*

Keywords recommender systems, multi-criteria analysis, user profiling

Citation Computer Science 17 (4) 2016: 503–517

1. Introduction

Recommender systems are a vital part of e-commerce; they are widely used in online shopping, advertising, and browsing. Algorithms used in such systems rarely include the general characteristics of users, such as psychological profiles. Nonetheless, the personality of human behavior may be easily added to the implemented algorithms to make more accurate recommendations. However, much research is still needed to give such an approach its final shape.

The main contributions of this paper are:

- Examination of Pareto Depth Analysis (PDA) for user classification
- Description of a new method to create categories of users
- Comparison of the proposed method's accuracy to the classical k-nearest neighbors (KNN) classification

The paper is organized as follows: Section 2 contains a definition and brief description of the algorithms used in recommender systems. It also provides some basic information on the quality of psychological questionnaires. Next, in Section 3, there is a general overview of multi-criteria optimization and its applications in the social sciences. This section also describes studies on how psychological data can be included in the recommender system. Section 4 characterizes the questionnaire-based dataset used in this study. In Section 5, the proposed classification method is introduced. Next, Section 6 provides more details on the implementation of the proposed method as well as the obtained results. The paper concludes with Section 7, where the discussion of the results and future work are described.

2. Background

2.1. Recommender system definition and basic algorithms

A recommender system is created to predict user preferences and behavior. It is exploited extensively in online retailing or browsing. The general model of all recommender systems is based on a *utility matrix* [18]. Its rows usually represent users, while its columns represent items (i.e., articles, products, etc.). Values for *user-item* pairs reflect the degree of preference a user shows towards a certain item. These values can be expressed on different measurement scales (i.e., 1-to-5 on the Likert scale, where 1 means disapproval and 5 means the strongest preference). The matrix is usually sparse; this means that certain user ranked only a few items. The main goal of a recommender system is to predict preference ratings for the undefined *user-item* pairs and suggest items that the user is expected to like. However, the prediction of user preference for each item does not always make sense. Therefore, choosing relevant *user-item* pairs is an important part of a recommender system [20]. A utility matrix can also be populated with boolean values, with 1's representing purchases(preferences) and 0's representing refrains from purchase.

Table 1 is an example of a utility matrix with user preference estimations. It can be clearly seen that the *User D* has rated his/her preference of Item 2 (I_2) as '1' and of Item 5 (I_5) as '3.' *User D* has not rated any other items.

Table 1

An example of a utility matrix with ratings.

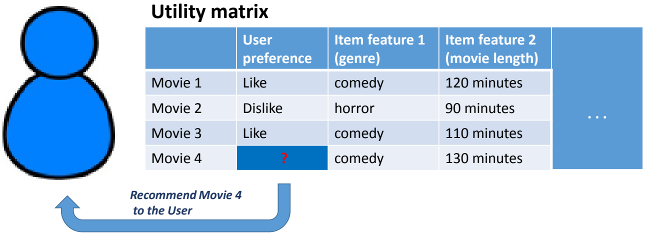
	I_1	I_2	I_3	I_4	I_5	I_6	I_7
User A	4		3	1			1
User B			4			5	
User C	2	2					5
User D		1			3		

A utility matrix can be populated by directly asking users to express their preferences, or they can be inferred using different premises. The latter may include previous the behavior of a user (to buy, watch, or read an item) and/or user general characteristics (i.e., demographics or psychological profile). Basic approaches underlying recommender systems include content-based filtering, collaborative filtering, or a combination of both [20]. Content-based recommender systems focus on the properties of the items selected by the user. Collaborative-filtering focuses on the similarities between users who selected a given item.

2.2. Content-based filtering

Content-based filtering relies on item features. In the case of movie recommendations, such features might include genre, director, cast members, etc. Users may also tag items to describe them. Features of an item are usually represented by a vector of 0's (if an item lacks a given feature) and 1's (if an item has a given feature). Numerical (not boolean) features can be directly represented in a vector. A collection of vectors describing a given item is called an *item profile.* Next, information about user preferences toward certain items from the utility matrix is analyzed. Recommendations are based on computing the *cosine similarity* between vectors of items and users (for details, see Sections 5.1.1). Normalization procedures should be introduced when using vectors with non-boolean values. The most-popular algorithm for grouping similar items and/or users profiles is *locality sensitive hashing* (LSH), which groups similar items into bins [12]. The algorithm suggests bins from which items should be recommended to a target user. Recommendations using items and user profiles can also be made by applying *decision trees* [20].

A simplified illustration of the general idea behind the content-based filtering is depicted in Figure 1. Knowing that Movie 4 is more similar to those a user previously liked (Movie 1 and Movie 3) than to one the user disliked (Movie 2), it can be recommended to a user.



Utility matrix

	User preference	Item feature 1 (genre)	Item feature 2 (movie length)	. . .
Movie 1	Like	comedy	120 minutes	
Movie 2	Dislike	horror	90 minutes	
Movie 3	Like	comedy	110 minutes	
Movie 4	?	comedy	130 minutes	

Figure 1. Simple scheme of content-based filtering.

2.3. Collaborative filtering

A collaborative-filtering approach to a recommender system focuses on similarities between users on ratings referring to the analyzed items. In the *model-based collaborative filtering*, a probabilistic model of user ratings is created to make predictions about the expected value of his/her future preferences [22]. This model is based mostly on a target user's history in the system rather than comparing him or her with other users in the system. Such probabilistic models are created using rule-based approaches, Bayesian networks [5], or clustering [4]. In memory-based collaborative filtering, users are grouped in sets depending on the ratings they share with the target user [4]. Users who share many ratings with the target user are called *neighbors*. The algorithm combines the neighbors' preferences to make inferences about the target user's future ratings.

A simple example of collaborative filtering is shown in Figure 2. Arrows reflect whether a user has seen a given item. The colors of the users represent their similarity. In this example, *User B* has seen two items in common with *User C* and is similar to him or her in terms of other characteristics. Therefore, *User C* will be presented items 1 and 2 (which he/she has not seen yet).

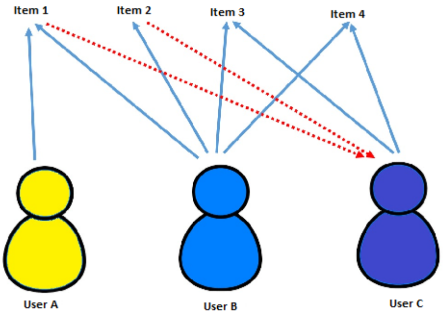


Figure 2. Simple scheme of content-based filtering.

2.4. Questionnaires construction

The most-intuitive way of measuring psychological traits is to use questionnaires. Psychological questionnaires are standardized sets of questions designed to measure psychological construct. A *score* is the sum of points assigned to answers given by the *testee* (a person filling out the questionnaire). It is assumed that the score from the psychological questionnaire reflects the intensity of the measured psychological construct. Higher scores represent a greater intensity of the measured trait. If a construct measured by a questionnaire is unidimensional, there is only one general score; however, if the measured psychological construct is multidimensional, questions referring to each dimension are grouped. Such a group is called a *psychological scale*. The score for each scale is calculated separately.

Each psychological questionnaire needs to meet certain quality standards. In other words, it needs to have specific *psychometric properties*. Such properties (like validity and reliability) always need to be proven in empirical studies [21]; therefore, professional questionnaires (both commercial and scientific) provide detailed data about the reproducibility of the results.

The reproducibility of the obtained results is measured with *reliability coefficients*. The most popular is Cronbachs alpha:

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k} s_i^2}{s_c^2}\right) \tag{1}$$

where:

k is the number of questions in a questionnaire,

s_c^2 is the variance of scores from all questions,

s_i^2 is the variance of scores from the analyzed psychological scale.

Cronbach's alpha coefficient informs us about the quality of a psychological questionnaire. It takes values from 0 to 1, where values above 0.7 are interpreted as sufficient for questionnaire use in an individual diagnosis.

3. Related work

3.1. Multi-criteria optimization

Multi-criteria optimization is a group of problems requiring the simultaneous optimization of more than one criterion m_j. Its main goal is to help decision makers compare possible alternatives by considering all defined criteria and choosing those most preferred. Let us assume that possible solutions X are vectors in a m-dimensional space, where m represents the analyzed criteria. Solution $x_i \epsilon X$, $X \epsilon R^m$ is a non-dominated (*Pareto optimal*) if no other $x_i \epsilon X$ is better or equal to x_i in all m criteria. The concept of better points in criteria space can be defined by maximizing, minimizing, or stabilizing criteria towards a certain value. A set of Pareto optimal solutions is called a Pareto front. To select among Pareto optimal solutions, diverse methods are used. A general method used for both finding Pareto optimal solutions and selecting among them is based on scalarizing functions; that is, scalar-valued functions of x_i that are optimized (maximized or minimized). Such functions might be linear or nonlinear, defined by weighting coefficients or by reference points in a criteria space. So far, multi-criteria analysis has been successfully applied to solve many real-life problems: in environmental protection, [26]; sociology, [7]; economics, [16]; and others. Moreover, multi-criteria decision-making methods have also been applied in Internet commerce [11], evaluating the quality of articles in Wikipedia [23], and optimizing multicast communication [27]. Another interesting application concerns the use of mutlicriteria methods in establishing fairness [25].

Assuming that the optimization problem is to minimize values on two criteria, a Pareto front in two-dimensional space is illustrated in a plot (Fig. 3). Points connected by a line form a Pareto front. Point C is not a part of the Pareto front, because it is dominated by point B (which takes smaller values on both the x_1 and x_2 criteria). Points A and B are not dominated. Point B takes smaller values on x_2 but greater values on x_1 than point A. Point A takes smaller values on x_1 but greater values on x_2 than point B. Points A and B are not strictly dominated by any other points.

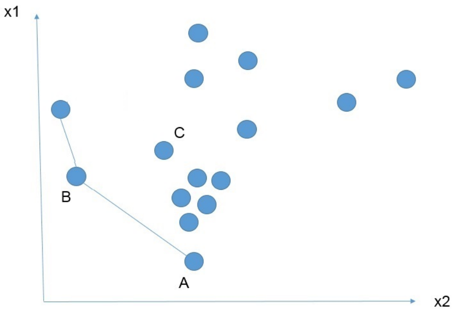


Figure 3. Illustration of a classification problem based on minimizing x_1 and x_2 criteria. Connected points form a Pareto front.

3.2. Including psychological factors in recommender systems

Attempts to personalize recommender systems by using psychological data about the users has recently become a popular research area. Taking into account a user's psychological profile is thought to enhance both his or her experience and the quality of recommendations. It has been proven that people with similar personality profiles are likely to share the same interests [19]. Therefore, incorporating personality profiles boosts the effectiveness of recommender system by increasing recommendation acceptance [3]. Examples of such successful implementations are described in [9, 17]. Another psychological variable that boosts the effectiveness of recommender systems is user mood state [13–15]. Results of a similar study conducted by Gonzales et.al. [6]

were implemented commercially (`http://emagister.com`). Gonzales proposed and developed the so-called Smart User Model (SUM) that uses emotional intelligence to adaptively create recommendations.

4. Dataset description

Dataset used for the analysis comes from the Polish normalization study of the Business-focused Inventory of Personality (BIP;[10]), conducted by the Polish Psychological Test Laboratory of the Polish Psychological Association. BIP is a psychological questionnaire used for candidate selection in recruitment procedures. The validity and reliability of this questionnaire prove its usefulness in professional and scientific applications. Cronbach's alpha in BIP oscillates around 0.8 for each scale. The inventory consists of 220 questions. Testees answer to each question is given a rank from a range of 1 to 6. The score obtained by each *testee* (person who fills out a questionnaire) is the sum of ranks assigned to his/her answers. In BIP, questions are grouped in 14 subsets (psychological *scales*). The score in each scale is the quantitative assessment of a distinct personality dimension.

Each observation in the dataset is represented by vector X of the scores obtained in each of the 14 scales (psychological profile):

$$X = [x_1, x_2, \ldots, x_m] \tag{2}$$

where m is the number of scales; $m = 14$.

Through statistical analysis, he inventory constructors distinguished four major *personality profiles* most characteristic for the representatives of certain professions. A personality profile is defined as a configuration of scores obtained in each scale of the inventory. The first profile described in [10] reflects the score characteristic for accountants. The second profile reflects BIP scores of the representatives of those professions that require establishing relationships with other people (i.e., HR specialists, nurses, psychologist, teachers). The third profile is characteristic of jobs that can be classified as dangerous (i.e., work in fire services, police, army). The fourth profile is most suitable for professions that require personnel management (i.e., managers, trainers).

Personality profiles are denoted as:

$$P_h = [\overline{x_{1h}}, \overline{x_{2h}}, \overline{x_{3h}}, \ldots, \overline{x_{mh}}] \tag{3}$$

where:

h is the number of personality profile; $h \epsilon \{1, 2, 3, 4\}$;
m is the number of criteria (psychological scales); $m = 14$;
$\overline{x}$ is the arithmetic mean score obtained in a scale.

In the standard application of the questionnaire, the testee's results are interpreted separately for each scale. The similarity to certain profiles might be assessed subjectively by the test administrator, but is not an element of the standard procedure of questionnaire-score interpretation. In this paper, an attempt to classify testees as one of four personality profiles P_h based on 14 scale scores X is described. The dataset contains responses to the BIP questionnaire of 646 representatives from the professorially active population of Poland. Each observation is classified to one of four personality profile categories. The ages of the respondents ranged from 20 to 65 y/o. The study concentrated on 18 professions. The data was gathered in accordance with the standardized procedure for psychological testing [2].

5. Proposed method

5.1. Mathematical apparatus

5.1.1. Calculating similarity in recommender systems

Calculating the similarity s between users or items for a recommender system can be defined in several ways. One is to calculate the cosine distance between vectors [20]. In the utility matrix, both users and items are represented by vectors in a m-dimensional space. Calculating the cosine of the angle between the vectors reflects similarity between them:

$$s(i,j) = \cos\left(\overrightarrow{i}, \overrightarrow{j}\right) = \frac{\overrightarrow{i} \cdot \overrightarrow{j}}{\|\overrightarrow{i}\| \, \|\overrightarrow{j}\|} \tag{4}$$

where $\cdot$ is a scalar-product of vectors i and j. Another way to measure similarity is to calculate correlation-based similarity. This can be done using a standard Pearson coefficient [20].

$$s(i,j) = \frac{\sum_{u \epsilon U} \left(R_{ui} - \overline{R_i}\right)\left(R_{uj} - \overline{R_j}\right)}{\sqrt{\left(R_{ui} - \overline{R_i}\right)^2} \sqrt{\left(R_{uj} - \overline{R_j}\right)^2}} \tag{5}$$

where:

U means users that have rated on the same items,

i, j are items that are being rated,

R_{ui} is the rating given by user u for item i.

5.1.2. Pareto-depth analysis (PDA)

This concept is a method based on a Pareto optimality. It determines a set of Pareto optimal points that form a Pareto front, eliminates them from the initial dataset, and searches for another Pareto front. This procedure is repeated until no more Pareto fronts can be determined. Hsiao et. al. [8] proposed using PDA for detecting outliers in the dataset. In his approach, the initial dataset is transformed into a set of *dyads*, which are pairwise comparisons of all observations from the initial dataset. As they reflect dissimilarities between pairs of observations. the scalarizing function in this approach is always a minimizing function. The PDA analysis is executed on the set of dyads. The final classification decision (*outlier-class member*) is a function of distance from the obtained Pareto fronts.

5.1.3. K – Nearest neighbors classification

K-nearest neighbors (KNN) is a non-parametric method commonly used for classification [1]. It assumes dividing the initial dataset into learning and testing subsets. All observations in the learning subset are assigned to a certain category (h). To classify an observation y from the testing subset, distances (usually Euclidean) between y and all observations from the learning subset are calculated. Next, k observations with the smallest distances from y are selected. The majority voting of the observations determines the classification decision of y. In other words, observation y is classified to the category h that is most common among k selected observations.

5.2. Classification based on the distance from Pareto fronts

In the paper, we propose a new method for user classification that exploits multi-criteria analysis. The general idea of the method is illustrated in Figures 4 and 5. Dots of different colors (green, yellow, and black) represent observations assigned to one of the classification groups. In the presented example, there are only three classification groups; however, the dataset used for the analyses contains four classification groups h representing psychological profiles.

The axes in Figure 4 represent dimensions of the psychological profile. For simplification in the example, only a two-dimensional space is presented. However, in the dataset used for the analyses, 14 dimensions were analyzed. Each dimension reflects the score obtained by a person in a scale from the BIP personality questionnaire.

The red dot represents a person that is to be classified to one of the groups.

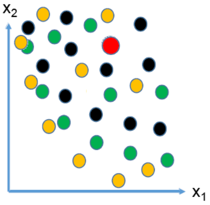


Figure 4. Illustration of a classification problem based on minimizing x_1 and x_2 criteria. Colors of the dots represent membership to one of three classes.

The first step in the proposed classification method (see Fig. 5) is to establish Pareto fronts for each analyzed group. In the second step, the distance between observations forming a Pareto front and target observation are calculated. The three smallest distances d from each Pareto front are then selected. In the third step, these values are used for target-user classification. The smallest distance from a Pareto front determines the assignment of a group membership to a target user.

The main idea of this paper is to include the distance from Pareto fronts as an additional feature in KNN classification. As described in Section 3.1, scalarizing functions used for creating Pareto fronts can take various forms – stabilizing, minimizing, or maximizing the analyzed criteria. Therefore, different methods for creating Pareto fronts are verified in the first step. Additionally, the application of Pareto-depth analysis (PDA) for the purpose of classification is examined. In the next step, the distance from a Pareto front established using best scalarizing function is used as an additional feature in the classical KNN classification of users. Accuracies of both methods are compared.

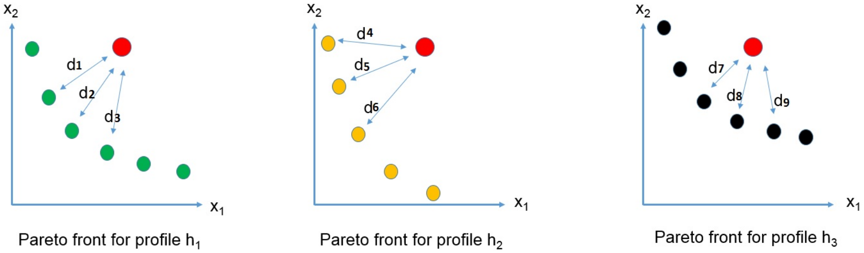

x2
x1
d1
d2
d3
Pareto front for profile h1
x2
x1
d4
d5
d6
Pareto front for profile h2
x2
x1
d7
d8
d9
Pareto front for profile h3

Figure 5. Illustration of the proposed classification method based on Pareto fronts. The red dot represents an object that needs to be classified. Green, yellow, and black dots represent Pareto front determined for groups h_1, h_2, and h_3 (see Section 4).

6. Computations and results

6.1. Cross-validation

Repeated Random Sub-Sampling (RRSSCV) methodology [24] was applied for cross-validation. The initial dataset was divided into training ($\boldsymbol{L}$) and testing ($\boldsymbol{T}$) subsets. The testing subset consisted of 50 randomly selected examples from the initial dataset. The learning dataset was constructed from the remaining observations. Each personality profile category h was equally represented in $\boldsymbol{L}$. Different sizes of $\boldsymbol{L}$ ranging from $4 \times 50 = 200$ to $4 \times 100 = 400$ were examined. This procedure was repeated 100 times. In each iteration, training was executed on $\boldsymbol{L}$ and the accuracy of classification was tested on $\boldsymbol{T}$. Mean classification accuracy for different sizes of $\boldsymbol{L}$ were verified.

6.2. Classification methods

The first classification method (C1) is based on the PDA anomaly detection proposed by Hsiao et.al. [8]. Computations were executed using the Matlab script available online. The implemented algorithm returns a standardized λ coefficient (for details, see [8]). Values close to 1 suggest that the observation is an outlier.

Observations from the testing set were compared to four $\boldsymbol{L}$ subsets representing h profiles defined by the BIP constructors (see Section 3). The final classification to one of four profile categories was made on the minimal value of λ.

In the second classification method (C2), a Pareto front for each profile group h in $\boldsymbol{L}$ was determined. Two types of scalarizing functions were examined:

- S1: stabilizing all x scores towards respective P_{mh} values,
- S2: mix of stabilizing, maximizing, and minimizing scores on different scales. If $\overline{x_{mh}}$ was greater than the third quartile of all scores, f was maximized. If $\overline{x_{mh}}$ smaller than the first quartile of all scores, f was minimized. In all other cases, f was stabilized towards the respective P_{mh} values.

Next, for the observations in $\boldsymbol{T}$, the three nearest neighbors (as defined in Section 6.3) from each h Pareto front were identified. Euclidean distance d_{ih} between these points and observations from the testing set were calculated. The final classification decision was made on the minimal sum of distances $min(\sum_{h=1}^{4} d_{ih})$.

Computations were executed using R CRAN software (version 3.1.2) with installed packages *FNN* and *rPref*. In both classification methods, classification accuracy and computation time were controlled for different sizes of training samples.

6.3. Distance from the Pareto front as a new feature in the standard classification method

The last step in the experiment was to include the distance from a Pareto front as an additional feature in the KNN classification of users. The standard KNN classification (treated here as a benchmark) uses only results obtained by users in 14 psychological questionnaire scales (see Section 4).

6.4. Results

The results of the classification accuracy for the C1 and C2 methods dependant on the size of the training sample are illustrated in Figure 6 followed by the evaluation of mean computation time shown in Figure 7. Classification method C1 proves to give poor results. With the growing size of the training sample, the classification accuracy increases; however, it never exceeds the level of 0.85 of the correct classification. The minimum average accuracy obtained in 100 iterations was 0.65. This result was observed for training sample size 50 × 4 = 200. The maximum classification accuracy of 0.84 was obtained for training sample size 99 × 4 = 396. Classification method C2 shows similar classification accuracy (oscillating around 0.85) irrespective of stabilization function or distance- calculation variants.

The mean classification accuracies based on the distance from the Pareto front (C2) are comparable and range between 0.83 to 0.86. For the next analysis, the Pareto front with stabilizing scalarizing function (C2–S1) was selected. The size of the training sample in further analyses was set at 400 observations.

The results of the classification accuracy are presented in Table 2. Three methods of classification were compared: distance from Pareto front (*PF classification*), classic KNN classification, and KNN classification with an additional feature reflecting the distance from Pareto front (*mixed classification*).

The results show that the accuracy of KNN classification is slightly better than PF classification. However, the combination of both methods gives the best results, with a mean accuracy of 0.89 in 100 iterations.

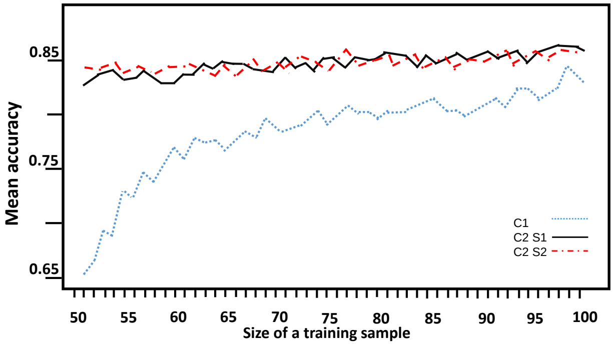


Figure 6. Mean classification accuracy in 100 iterations for different training sample sizes (description as in Section 6.2).

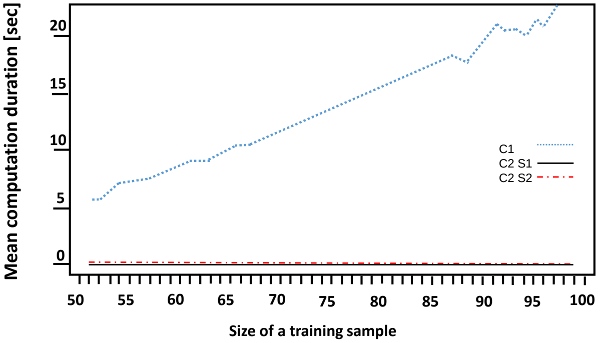


Figure 7. Mean computation duration in 100 iterations for different training sample sizes (description as in Section 6.2).

Table 2

Classification accuracy for analyzed classification methods (100 iterations).

Classification approach	Mean	Max
PF classification	0.85	0.87
KNN classification	0.86	0.88
mixed classification	0.89	0.93

7. Discussion and future work

Experimental results show that the PDA-based classification method modeled on the Hsiaos [20] algorithm gives unsatisfactory results, both in terms of classification accuracy and computation duration. However, the second classification method examined in this paper that uses Pareto fronts provides acceptable accuracy (around 0.85) and computation duration. This result is observed irrespective of the variations in scalarizing function and distance calculation. The PDA-based algorithm [20] based on the *dyads* reflect pairwise differences between observations on all criteria. In this case, the scalarizing function does not require us to structure the criteria using expert opinions or established in empirical studies. The algorithm simply searches for different layers of Pareto fronts by minimizing the differences between the dyads. The obtained results might suggest that information from the human expert (or empirical studies) are still difficult to automatically replace in multi-criteria classification.

Moreover, results described in the initial Hsiao paper [8] described binary classification (*outlier-class member*) computed on simulated data. Perhaps the proposed algorithm (applied to empirical data and faced with more-complex classification) needs modifications in order to produce satisfactory classification accuracy.

When it comes to the proposed method of classification based on distances from Pareto fronts, the results look promising. Despite the fact that the mean accuracy of this classification method is similar to the accuracy of the classical KNN method, a combination of these two approaches results in a mean increase in accuracy.

In the future, we are planning to check how the application of the proposed method in collaborative filtering-based recommender system affects its effectiveness and recommendation acceptance. We would also like to examine other types of distances like L^1 and L^∞ and check how they influence the obtained results. Moreover, it would be interesting to check how other methods for splitting the dataset into learning and testing samples modify the outcome of the experiment.

References

[1] Altman N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, vol. 46(3), pp. 175–185, 1992.

[2] American Educational Research Association, American Psychological Association, National Council on Measurement in Education: *Standards for educational and psychological testing*. American Educational Research Association, 1999, `https://books.google.pl/books?id=BMOQAQAAMAAJ`.

[3] Bonhard P., Harries C., McCarthy J., Sasse M.A.: Accounting for taste: using profile similarity to improve recommender systems. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1057–1066, ACM, 2006.

[4] Breese J.S., Heckerman D., Kadie C.: Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43–52, Morgan Kaufmann Publishers Inc., 1998.

[5] Condliff M.K., Lewis D.D., Madigan D., Posse C.: Bayesian mixed-effects models for recommender systems. In: *ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*, vol. 15, Citeseer, 1999.

[6] González G., López B., de la Rosa J.L.: A multi-agent smart user model for cross-domain recommender systems. *Proceedings of Beyond Personalization*, 2005.

[7] Gross M., Heinrichs H.: *Environmental sociology: European perspectives and interdisciplinary challenges.* Springer Science & Business Media, 2010.

[8] Hsiao K.J., Xu K., Calder J., Hero A.O.: Multi-criteria anomaly detection using Pareto Depth Analysis. In: *Advances in Neural Information Processing Systems*, pp. 845–853, 2012.

[9] Hu R., Pu P.: A study on user perception of personality-based recommender systems. In: *User Modeling, Adaptation, and Personalization*, pp. 291–302, Springer, 2010.

[10] Jaworowska A., Brzezińska U.: *BIP Bochumski Inwentarz Osobowościowych Wyznaczników Pracy.* Pracownia Testów Psychologicznych Polskiego Towarzystwa Psychologicznego, 2014.

[11] Kaszuba T., Hupa A., Wierzbicki A.: Advanced feedback management for internet auction reputation systems. *Internet Computing, IEEE*, vol. 14(5), pp. 31–37, 2010.

[12] Lops P., De Gemmis M., Semeraro G.: Content-based recommender systems: State of the art and trends. In: *Recommender systems handbook*, pp. 73–105, Springer, 2011.

[13] Masthoff J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. In: *Personalized Digital Television*, pp. 93–141, Springer, 2004.

[14] Masthoff J.: The pursuit of satisfaction: affective state in group recommender systems. In: *User Modeling 2005*, pp. 297–306, Springer, 2005.

[15] Masthoff J., Gatt A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, vol. 16(3–4), pp. 281–319, 2006.

[16] Munda G.: *Social multi-criteria evaluation for a sustainable economy.* Springer, 2008.

[17] Nunes M.A.S.N.: *Recommender systems based on personality traits.* Ph.D. thesis, Université Montpellier II Sciences et Techniques du Languedoc, 2008.

[18] Rajaraman A., Ullman J.D.: *Mining of massive datasets*, vol. 77. Cambridge, University Press Cambridge, 2012.

[19] Rentfrow P.J., Gosling S.D.: The do re mi's of everyday life: the structure and personality correlates of music preferences. *Journal of personality and social psychology*, vol. 84(6), p. 1236, 2003.

[20] Ricci F., Rokach L., Shapira B.: *Introduction to recommender systems handbook.* Springer, 2011.

[21] Rust J., Golombok S.: *Modern psychometrics: The science of psychological assessment.* Routledge, 2014.

[22] Sarwar B., Karypis G., Konstan J., Riedl J.: Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001.

[23] Turek P., Wierzbicki A., Nielek R., Hupa A., Datta A.: Learning about the quality of teamwork from wikiteams. In: *Social Computing (SocialCom), 2010 IEEE Second International Conference*, pp. 17–24, IEEE, 2010.

[24] Ward R.: Compressed sensing with cross validation. *Information Theory, IEEE Transactions on*, vol. 55(12), pp. 5773–5782, 2009.

[25] Wierzbicki A.: The case for fairness of trust management. *Electronic Notes in Theoretical Computer Science*, vol. 197(2), pp. 73–89, 2008.

[26] Wierzbicki A., Makowski M., Wessels J.: *Model-based decision support methodology with environmental applications.* Kluwer Academic Dordrecht, The Netherlands, 2000.

[27] Wierzbicki A., Szczepaniak R., Buszka M.: Application layer multicast for efficient peer-to-peer applications. In: *Internet Applications. Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003*, pp. 126–130, IEEE, 2003.

Affiliations

Maria Rafalak
Polish-Japanese Academy of Information Technology, Warsaw Koszykowa 86,
m.rafalak@pjatk.edu.pl

Janusz Granat
National Institute of Telecomunications, Warsaw Szachowa 1, j.granat@itl.waw.pl

Andrzej P. Wierzbicki
National Institute of Telecomunications, Warsaw Szachowa 1, a.wierzbicki@itl.waw.pl

Received: 31.01.2016
Revised: 16.05.2016
Accepted: 17.05.2016