On the quantification of missing value impact on Voting Advice Applications

Marilena Agathokleous, Nicolas Tsapatsoulis and Ioannis Katakis

30, Arch. Kyprianos str., CY-3036, Limassol, Cyprus
{mi.agathokleous@edu.cut.ac.cy},{nicolas.tsapatsoulis@cut.ac.cy},
{ioannis.katakis@gmail.com}
http://nicolast.tiddlyspot.com

Abstract. Voting Advice Application (VAA) is a web application that recommends a candidate or a party to a voter. From an online questionnaire, which voters and candidates are called to answer, the VAA proposes to each individual voter the candidate who replied like him/her. It is very important the voters to reply in all questions of the questionnaire, because every question has its meaning and is responding to the political position of a each party. Missing values might mislead the VAA and impede it to have complete knowledge about the voter, as a result to offer him/her the wrong candidate. In this paper we quantitatively investigate the effect of missing values in VAAs by examining the impact of the number of missing values to different methods of voting prediction. For our experiment we have used the data obtained from the May parliamentary elections in Greece in 2012. The corresponding dataset is made freely available to other researchers working in the areas of VAA and recommender systems through the Web.

Key words: Missing values, classifiers, recommender systems, voting advice applications

1 Introduction

The Voting Advice Application (VAA), is an online survey tool that has recently been successfully used in many European countries and lately in countries outside the European continent [8]. It has been characterized as a new phenomenon in modern election campaigning and its utilization has increased in recent years [2]. VAAs' target is to help voters identify parties that have similar political positions with themselves. Also VAAs encourage, indirectly, citizens to exercise their right to vote and to be informed about the political stances of parties.

The operation of a VAA is briefly summarized as follows: First, parties (through representatives) are called to answer a set of questions in an online questionnaire. This questionnaire typically consists of a set of policy statements on which the parties' positions have been coded. Each question corresponds to the political positions of the parties and their reaction to the developments in the current affairs. Preferred questions are those encoding a high variance in party positions. Once the party answers on the questionnaire were encoded citizens are able to fill in the same questionnaire by navigating to the VAA website.Both voters and candidates (party representatives) evaluate each issue by giving lower extent to those with which they do not agree at all and higher to those that perfectly expressed their position [2][6]. Usually the answering options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree' and 'I have no opinion'. In the end, the similarity between voters and candidates is calculated, and, with the aid of a properly designed algorithm every voter is recommended the candidate with whom he/she has the higher similarity. In the following sections of this paper this, traditional, method of voting recommendation will be referred as 'Party Coding'.

Recently, Katakis *et al.* [5] proposed an alternative method of voting recommendation in VAAs in an effort to extend their social dimension and transform them into the so-called Social VAAs (SVAAs). In SVAAs in addition to the traditional recommendation which is based on party-voter similarity, voters are recommended candidates using a collaborative filtering perspective [10] by considering SVAAs a special type of recommender systems [9]. The overall idea behind this is that voters are recommended candidates based on their similarity with other voters who expressed their voting intention. In most VAAs the voting intention of the users is recorded by asking them a corresponding complimentary question (see `www.choose4greece.com` and `www.choose4cyprus.com`). On average, 70% of the users answer this question while about 50% of them express voting intention for a specific party or candidate. Katakis gets advantage of these data and proposes several voting prediction schemes by modeling party voters using machine learning tools. He showed that party voter modeling methods are superior to party coding as far as the voting prediction is concerned and the inclusion of this type of prediction in VAAs, as an additional component, increases the enjoyment of VAA users.

Despite the recommendation method (party coding or party voter modeling) the recommendation given to the voters must be accurate. This happens whenever citizens reply in all questions of the online questionnaire, without leaving missing values. Similarly the information about candidates' or party positions is required to be more wide, which will be fact if the candidates do not leave missing values too. Unfortunately, in may times the candidates refrain from exposing themselves to controversial issues, choosing answers in the middle of the Likert scale (i.e., 'neither agree nor disagree') while the voters leave unanswered several questions or give answers like 'I have no opinion' or 'neither agree nor disagree' for several reasons, including time constraints, limited information on the corresponding issues, or even due to unclear questions. Furthermore, sometimes voters forget or avoid answering a question. Also there are questions that are characterized of ambivalence or indecisiveness and those with which the respondent does not agree against the main assumptions of them [1]. These values can be characterized as missing values and, in theory, it is considered that they seriously affect VAAs voting recommendation.

In this paper we try to quantify the effect of missing values on VAAs voting recommendation effectiveness both for the traditional party coding method as well as for the party voter modeling methods. By gradually increasing, randomly, the number of missing values we measure the average precision and recall values for the various VAA recommendation methods. In this way we identify in practice the robustness of each recommendation method while we prove empirically to which extent missing values affect VAAs performance. To the best of our knowledge it is the first time that both missing values' impact quantification and recommendation method robustness are dealt with. Conclusions on these two aspects are important because they seriously affect VAAs design and VAA data utilization. The experiments were conducted on the VAA dataset obtained from Greek parliamentary elections of May, 2012, which is online available at `www.choose4greece.com/datasets`.

2 Problem formulation

The basic aim of a VAA is to recommend parties of candidates to users. In such a case there is a set of N users $U = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_N}\}$, a set of M questions (or issues) $Q = \{q_1, q_2, \ldots, q_M\}$, and a set of T political parties (or candidates) $P = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_T}\}$. Each user $\boldsymbol{u_i} \in U$ and each political party $\boldsymbol{p_j} \in P$, has answered each question $q_k \in Q$. The answers of users are recorded through online questionnaires like the one in Choose4Greece (`www.choose4greece.com`). The answers of political parties are either coded by experts or answered by representatives of political parties.

Based on their answers, every political party or user can be represented in a vector space model:

$$\boldsymbol{u_i} = \{u_{(i,1)}, u_{(i.2)}, \ldots, u_{(i,k)}, \ldots, u_{(i,M)}\} \tag{1}$$

$$\boldsymbol{p_j} = \{p_{(j,1)}, p_{(j,2)}, \ldots, p_{(j,k)}, \ldots, p_{(j,M)}\} \tag{2}$$

where $u_{(i,k)}, p_{(j,k)} \in L$ are the answers of the i-th user and j-th party, respectively, to the k-th question. Usually, vectors $\boldsymbol{u_i}$ and $\boldsymbol{p_j}$ are named *profiles*.

A typical set of answers is a 6-point Likert scale: L ={-2 (Strongly disagree), -1 (Disagree), 0 (Neither agree nor disagree), 1 (Agree), 2 (Strongly agree), 3 (No opinion)} but in practice the sixth point it is not taken into consideration since does not correspond to a particular stance. As a result the set L, in the context of this work, becomes: L ={-2,-1,0,1,2}.

The VAA recommendation *task* can be then defined as follows: Given the answers of a specific user $\boldsymbol{u_a}$ suggest a ranking of political parties based on user-party relevance. In machine learning terms, the task is to approximate the hidden function $h : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$, where $h(\boldsymbol{u}, \boldsymbol{p})$ is the estimation of the relevance of user $\boldsymbol{u}$ with political party $\boldsymbol{p}$. Typically $h(\boldsymbol{u}, \boldsymbol{p}) \in [0, 1]$. In each case, the top suggestion p_a for user u_a should be:

$$p_a = \underbrace{argmax}_{j}(h(\boldsymbol{u_a}, \boldsymbol{p_j})) \tag{3}$$

Similarly, we could consider a function $r(\boldsymbol{u},\boldsymbol{p}) \in [1,T]$ that returns the *rank* of the political party p for the user u, if all political parties are ranked according to relevance (similarity) with this specific user. Having learned function $h(\boldsymbol{u},\boldsymbol{p})$ it is straightforward to calculate $r(\boldsymbol{u},\boldsymbol{p})$.

In order to produce vote recommendations, the most simple approach is to estimate $h(\boldsymbol{u},\boldsymbol{p})$ with the aid of a distance measure $d(\boldsymbol{u},\boldsymbol{p})$. A number of such distance measures are discussed in [7]. In this case the top suggestion p_a for user u_a is given by:

$$p_a = \underbrace{argmin}_{j}(d(\boldsymbol{u_a},\boldsymbol{p_j})) \tag{4}$$

The above mentioned recommendation approach is the one, traditionally, used in VAAs and in the context of this paper is referred to as 'Party Coding' method.

In many voting assistance systems, the information of *vote intention* v_i of many users $\boldsymbol{u_i}$ is available as it is included as a supplementary question in the online surveys. This kind of information can be utilized to model party voters using statistical or machine learning approaches and provide an additional kind of recommendation that is based on collaborative filtering (see [5]). These type of approaches are referred to, in the context of this work, as 'Party voter modeling' methods.

In the framework described in this section a missing value is considered every $u_{(i,k)}, p_{(j,k)} \notin L$, that is, any question k left unanswered or given the value 'I have no opinion' by either a particular user i or a party representative (or candidate) j. In order to quantify the impact of the number of missing values on the VAA recommendation we a need a few more definitions.

Let us define the utility matrix U as an NxM matrix whose rows are the transposed user vectors $\boldsymbol{u_i}^T$, $i = 1 \dots N$. That is:

$$U = \begin{pmatrix} u_{(1,1)} & u_{(1,2)} & \cdots & u_{(1,M)} \\ u_{(2,1)} & u_{(2,2)} & \cdots & u_{(2,M)} \\ \cdots & \cdots & \ddots & \cdots \\ u_{(N,1)} & u_{(N,2)} & \cdots & u_{(N,M)} \end{pmatrix} \tag{5}$$

The sparsity S_U of matrix U is defined as the percentage of missing values in U. That is:

$$S_U = \frac{|O|}{N \cdot M} \tag{6}$$

where $O = \{u_{(i,k)} | u_{(i,k)} \notin L, i = 1 \dots N, k = 1 \dots M\}$ is the set of missing values and $|O|$ is its cardinality.

The impact of missing values in U to VAA recommendations is quantified with the aid of well-known measures defined in information retrieval. In particular Precision, Recall and F-measure are computed for all voters of a particular party and a weighted average is calculated. Let us defined D_q the set of voters that expressed a voting intention for party p_q, that is:

$$D_q = \{u_i | u_i : v_i = p_q\} \tag{7}$$

Let us also define as D the set of users who expressed a voting intention for a particular party (i.e., the set of users that answer the corresponding complimentary question in VAA):

$$D = \bigcup_{q=1:T} D_q \tag{8}$$

We also define the set T_q as the set of users who expressed a voting intention for party p_q and the VAA recommendation coincides with their voting intention, and F_q the set of users who expressed a voting intention for a party different than p_q but the VAA recommended them p_q, i.e.,

$$T_q = \{u_i | u_i : v_i = p_q, p_q = \underbrace{argmax}_{j}(h(\boldsymbol{u_i}, \boldsymbol{p_j}))\} \tag{9}$$

$$F_q = \{u_i | u_i : v_i \neq p_q, p_q = \underbrace{argmax}_{j}(h(\boldsymbol{u_i}, \boldsymbol{p_j}))\} \tag{10}$$

With the aid of the definitions above we can formally define the per party Precision (Pr^q) and Recall (Re^q) measures as follows:

$$Pr^q = \frac{|T_q|}{|T_q| + |F_q|} \tag{11}$$

$$Re^q = \frac{|T_q|}{|D_q|} \tag{12}$$

where $|A|$ denotes the cardinality of set A.

The F-measure for a particular part p_q is defined as usual with the aid of Pr^q and Re^q:

$$Fm^q = \frac{2 \cdot Pr^q \cdot Re^q}{Pr^q + Re^q} \tag{13}$$

The overall Precision (Pr), Recall (Re) and F-measure (Fm) are computed as weighted sums of the per party corresponding quantities:

$$Pr = \frac{1}{|D|} \sum_{q=1:T} (|Dq| \cdot Pr^q) \tag{14}$$

$$Re = \frac{1}{|D|} \sum_{q=1:T} (|Dq| \cdot Re^q) \tag{15}$$

$$Fm = \frac{1}{|D|} \sum_{q=1:T} (|Dq| \cdot Fm^q) \tag{16}$$

It can be easily proved that the overall Recall (Re), computed as indicated above, coincides with the well-known accuracy measure (for a definition of the accuracy measure in VAAs please see [5]).

3 Methodology

Our methodology for the quantification of missing value impact on VAAs recommendation is quite simple: The number of missing values in the utility matrix U (see eq. (5)) are progressively increased by randomly removing matrix entries and the performance of each one of the recommendation methods, mentioned next, is evaluated with the aid of Precision, Recall and F-measure defined in equations (14)-(16). The random selection of matrix entries that are removed guarantees that no specific patterns will be created within matrix U.

The following voting recommendation methods were compared using the dataset collected from `www.choose4greece.com`. In its initial form it includes 75,294 voters. However, only 26,355 voters expressed voting intention for a particular party. Thus, in order to be able to apply the proposed methodology for the quantification of missing value impact on VAA recommendation we have used this particular subset of the original dataset. The original sparsity (see eq. (6)) of this dataset is negligible (0.0184).

Party Coding: Party coding is the traditional VAA recommendation method. The user u_a is recommended party p_a according to eq. (4). Although alternative distance metrics were proposed for VAAs [7] in this article we stick in, for the sake of simplicity, to the classic Euclidean distance for our experiments.

$$d(\boldsymbol{u_a}, \boldsymbol{p_j}) = \sqrt{\sum_{k=1}^{M} (u_{a,k} - p_{j,k})^2} \tag{17}$$

Party coding is by far the most simple and computationally undemanding method. Furthermore, is the only method that serves the second basic purpose of VAAs: provide information about the party positions.

Average Voter: Average Voter is a simple community-based approach for voting recommendation. It also makes use of eq. (4) with the aid of Euclidean distance. However, the difference is that instead of using the actual profile $\boldsymbol{p_j}$ of party p_j it utilizes the averaged profile $a(\boldsymbol{p_j})$ of all users that belong to set D_j:

$$a(\boldsymbol{p_j}) = \frac{1}{|D_j|} \sum_{u_i \in D_j} \boldsymbol{u_i} \tag{18}$$

The advantages of this approach are its simplicity and the fact that it does not need the profile of each political party, since it depends on the voters' answers. On the other hand this can be also its main disadvantage, as it is necessary to have a sufficient number of users that were expressed voting intention for party p_j in order to estimate a representative profile for the average voter.

Mahalanobis Classifier: Mahalanobis Classifier extends the idea of the average voter by taking into account two more pieces of information contained in the covariance matrix C_j of the voters of party p_j. It gives different weights to the various questions with the aid of the diagonal elements of matrix C_j while it records the correlations between questions through the non-diagonal elements.

The Mahalanobis Classifier makes use of eq. (3) where the relevance $h(\boldsymbol{u_a}, \boldsymbol{p_j})$ of user u_a with political party p_j is computed through the following equation:

$$h(\boldsymbol{u_a}, \boldsymbol{p_j}) = e^{-\frac{1}{2}\left((\boldsymbol{u_a} - a(\boldsymbol{p_j}))^T \cdot \ C_j^{-1} \cdot (\boldsymbol{u_a} - a(\boldsymbol{p_j}))\right)} \tag{19}$$

where $a(\boldsymbol{p_j})$ is the average profile of users belonging to set D_j and C_j is the corresponding covariance matrix given by:

$$C_j = \frac{1}{|D_j|} \sum_{u_i \in D_j} (\boldsymbol{u_i} - a(\boldsymbol{p_j})) \cdot (\boldsymbol{u_i} - a(\boldsymbol{p_j}))^T \tag{20}$$

Machine Learning Approaches: If we ignore the information of political party profiles, then the problem can be defined as a single-class data classification problem, with the class obviously being the vote intention of the user. The data matrix ($D_{N \times M}$) consists of all the users profiles and the class (label) vector

$$V_{N \times 1} = \{\nu_1, \ldots, \nu_N\}^T \tag{21}$$

consists of the vote intentions of all N users. Hence, $D_{N \times M}$ and $V_{N \times 1}$ constitute the training *examples* of the learning problem. In this case many classifiers can be represented as score functions $f(\boldsymbol{x}, y)$ that output the probability that instance $\boldsymbol{x}$ belongs to class y (i.e. $P(y|\boldsymbol{x})$), and $\sum_{y=1}^{T} f(\boldsymbol{x}, y) = 1$. Hence, in order to solve the VAA problem, it is straightforward to set $h(\boldsymbol{u}, p) = f(\boldsymbol{x}, y)$ and use as f one of the variety of classifiers (Decision Trees Classifiers, Bayesian Classifiers, Support Vector Machines, Neural Networks, etc). In essence what is achieved in this case is the modeling of the voters' behavior based on the questions.

We used the Weka software package [11] to train a variety of indicative classifiers using the 'one against the rest' training approach. Datasets were split into training and test sets, with a percentage of 66 percent and 34 percent respectively. We have implemented party voter models using the Naive Bayes classifier (bayes.NaiveBayes), the Logistic Regression classifier (functions.Logistic) and the Instance-Based classifier with fixed neighborhood (lazy.IBk) in order to cover statistical learning, neural networks and decision trees respectively. All these classifiers were set with Weka default parameter settings [3].

4 Experimental Results and Discussion

Experiments were designed to investigate the performance on voting recommendation of the classifiers, mentioned in the previous section, as a function of the number of missing values in the utility matrix U. Tables 1-3 summarize these

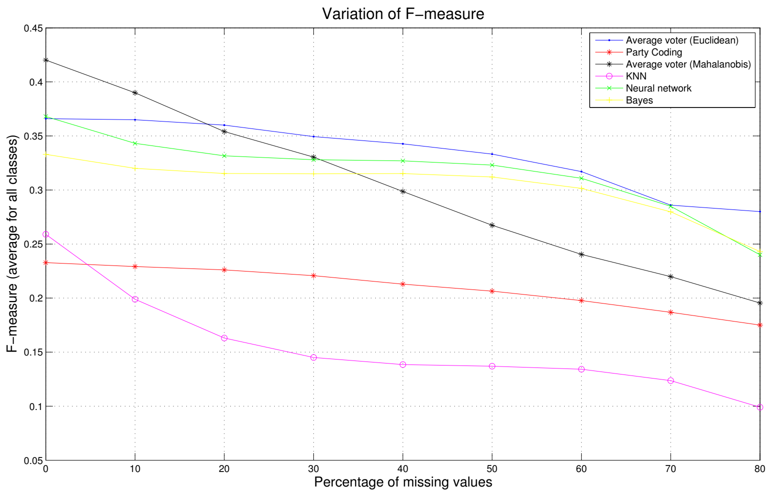

Fig. 1. Variation of F-measure as a function of the number of missing values

Table 1. Voting prediction evaluation on the original Choose4Greece dataset.

Classifier	Precision (average)	Recall (average)	F-measure (average)
Party Coding	0.268	0.206	0.233
Average Voter	0.427	0.328	0.371
Mahalanobis Classifier	0.424	0.417	0.420
KNN (lazy.IBk)	0.258	0.260	0.259
Neural Network (functions.Logistic)	0.357	0.399	0.368
Bayes (bayes.NaiveBayes)	0.328	0.343	0.333

Table 2. Voting prediction evaluation: Sparsity 30%.

	Sparsity 30%		
Classifier	Precision	Recall	F-measure
Party Coding	0.265	0.189	0.221
Average Voter	0.423	0.297	0.349
Mahalanobis Classifier	0.364	0.302	0.330
KNN (lazy.IBk)	0.168	0.135	0.145
Neural Network (functions.Logistic)	0.319	0.366	0.328
Bayes (bayes.NaiveBayes)	0.310	0.334	0.315

Table 3. Voting prediction evaluation: Sparsity 50%.

	Sparsity 50%		
Classifier	Precision	Recall	F-measure
Party Coding	0.262	0.170	0.207
Average Voter	0.421	0.276	0.343
Mahalanobis Classifier	0.321	0.229	0.267
KNN (lazy.IBk)	0.154	0.126	0.137
Neural Network (functions.Logistic)	0.299	0.350	0.323
Bayes (bayes.NaiveBayes)	0.308	0.331	0.312

results on four distinct sparsity values: 0%, 30%, and 50% respectively. Figure 1 on the other hand illustrates graphically the overall F-measure (see eq.(16)) as a function of the number of missing values.

It can be easily seen in the results that as the number of missing values increases the performance of all classifiers decreases. However, the rate of decrease varies. While the performance of Mahalanobis classifier and decision trees drops rapidly the other classifiers deteriorate gracefully. Overall the Bayesian classifier appears to be the more robust while the Neural Network classifier shows the highest performance in low to medium number of missing values. It is also interesting to note that even with a sparsity level as high as 50% all classifiers operate far above chance level (which in our case is 100/15 because the number of parties T is 15). Furthermore, sparsity levels higher than 10% is quite unusual in properly designed VAAs. Thus, it seems that the impact of missing values on VAA recommendation is somehow overestimated. Nevertheless, proper methods for missing value estimation are always useful and welcome.

5 Conclusion

In this article we investigated the impact of missing values on VAAs recommendation. We compared the traditional party coding recommendation method with several party voter modeling methods as far as the robustness to the number of missing values is concerned. The results show that in all cases the higher the number of missing values the lowest the recommendation accuracy (measured as per class recall in this study). However, the deterioration of performance is not as high as one might assume. Even if 80% of the expected data are missing the recommendation accuracy drops less than 50% of its highest value, showing a remarkable robustness. Among the various compared recommendation methods the traditional party coding and the ones that are based on Neural Networks and Bayesian inferencing are the more robust. The most affected methods by missing values is the one that is based on the covariance matrix of party voters (Mahalanobis Classifier) and the Decision Trees. Both are strongly related on the correlation between the questions of the questionnaire and as the number of missing values increases modeling this correlation becomes more difficult.

An implicit assumption we have made in this study is that missing values are replaced by neutral answers ('neither agree nor disagree'). However, this is not actually the case in practice because gives an advantage to parties or candidates that avoid to express specific positions in highly controversial issues. As a result in many VAAs the similarity between candidates and voters is measured using variations of the Euclidean distance and excluding the values in the middle scale (see [7]). We are currently investigating the impact of replacing missing values with alternate values outside the scale used in VAAs. That is, if we assume the value '-2' for 'strongly disagree' and the value '2' for 'strongly agree' then half of the missing values will be given the value '-3' and the other half the value '3'. Actually, some machine learning tools such as the Weka (which we have used in this study) accept missing values as inputs. However, it seems that internally they handle those values as neutral ones.

References

1. Baka, A., Figgou, L., Triga,V.: 'Neither agree, nor disagree': a critical analysis of the middle answer category in Voting Advice Applications. Int. J. Electronic Governance, vol. 5, no. 3/4, pp. 244-263 (2012)
2. Cedroni, L., Diego, G. (Eds.): Voting Advice Applications in Europe: The State of the Art. ScriptaWeb, Napoli (2010)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The weka data mining software: an update, SIGKDD Explor. Newsl., vol. 11, no. 1, pp. 10-18, (2009)
4. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd edition (2011)
5. Katakis, I., Tsapatsoulis, N., Triga, V., Tziouvas, C., Mendez, F.: Clustering Online Poll Data: Towards a Voting Assistance System. In: Semantic and Social Media Adaptation and Personalization (SMAP'12), IEEE Press, pp. 54-59 (2012)
6. Ladner, A., Pianzola, J.: Do voting advice applications have an effect on electoral participation and voter turnout; Evidence from the 2007 swiss federal elections. In Tambouris, E., Macintosh, A., Glassey, O. (eds) Electronic Participation. LNCS, vol. 6229, pp 211-224. Springer Berlin Heidelberg (2010)
7. Mendez, F.: Matching voters with political parties and candidates: An empirical test of four algorithms. International Journal of ElectronicGovernance, (2012)
8. Pianzola, J., Trechsel, A. H., Schwerdt, G., Vassil, K., Alvarez, R. M.: The Effect of Voting Advice Applications (VAAs) on Political Preferences. Evidence from a Randomized Field Experiment. In: Annual Meeting of the American Political Science Association (2012). Available at SSRN: `http://ssrn.com/abstract=2108095`
9. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.): Recommender Systems Handbook, pp.39-71. Springer New York Dordrecht Heidelberg London (2011)
10. Tsapatsoulis, N., Georgiou, O.: Investigating the Scalability of Algorithms, the Role of Similarity Metric and the List of Suggested Items Construction Scheme in Recommender Systems. International Journal on Artificial Intelligence Tools, vol. 21, no. 4, pp. 12-40 (2012)
11. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2nd edition, pp. 3-9 (2005)