Voting Advice Applications: Missing Value Estimation Using Matrix Factorization and Collaborative Filtering

Marilena Agathokleous, Nicolas Tsapatsoulis

HAL Id: hal-01459677
https://inria.hal.science/hal-01459677

Submitted on 7 Feb 2017

Voting Advice Applications: Missing value estimation using matrix factorization and collaborative filtering

Marilena Agathokleous and Nicolas Tsapatsoulis

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

Abstract. A Voting Advice Application (VAA) is a web application that recommends to a voter the party or the candidate, who replied like him/her in an online questionnaire. Every question is responding to the political positions of each party. If the voter fails to answer some questions, it is likely the VAA to offer him/her the wrong candidate. Therefore, it is necessary to inspect the missing data (not answered questions) and try to estimate them. In this paper we formulate the VAA missing value problem and investigate several different approaches of collaborative filtering to tackle it. The evaluation of the proposed approaches was done by using the data obtained from the Cypriot presidential elections of February 2013 and the parliamentary elections in Greece in May, 2012. The corresponding datasets are made freely available to other researchers working in the areas of VAA and recommender systems through the Web.

Key words: Missing values, collaborative filtering, recommender systems, voting advice applications

1 Introduction

Democracy is the form of government where power emanates from the people, exercised by the people and serves the interests of the people. A central feature of democracy is the decision by voting citizens, in direct democracy, or some representatives, in a representative democracy [16]. However, a fairly large percentage of citizens are accustomed to not exercise their right to vote and many of them are not even informed of the political positions of the candidates. A voting advice application (VAA) is a web application that helps voters to be informed about the political stances of parties and to realize with which of them are closest. VAAs are a new phenomenon in modern election campaigning, which has increased in recent years [5].

Voting Advice Applications pose voters and candidates to answer a set of questions in an online questionnaire. Each question corresponds to the political positions of the parties and their reaction to the developments in the current affairs. Voters and candidates 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 [5][12]. 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.

VAAs are becoming increasingly common in electoral campaigns in Europe. A survey in 2008 showed that VAAs were applied to general elections in 15 European countries, having high impact in most cases [21]. Many argue that VAAs are able to increase and improve democratic participation, since these applications help voters to have easy access in information about the political views of political parties. For better utilization of such applications is essential that advices by VAAs be credible and impartial. Additionally, it is desirable to have access to a broad cross section of society, not only by people of one side of the political and social spectrum.

In VAAs it is highly desirable that both the users and the candidates answer all the questions of the online questionnaire. In this case the recommendation given to the voter is more accurate and the information about candidates' positions are more wide. Unfortunately, this is not the case. 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. 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 [17].

Althought missing values distort , to some extent, the VAAs. Choosing to ignore them might severely affect the overall sample and it is likely to provide a completely wrong result: it is highly probable the VAA to not be able to estimate the similarity between the voter and the candidates and suggest a wrong candidate to a voter; even if the missing data are not a extended (only few questions left unanswered). As a result the effectiveness and reliability of the corresponding VAA will be deteriorated. In this paper we formulate missing values in VAAs as a recommender system problem and we applied state of the art techniques from this discipline to effectively tackle it. In particular, we investigate both matrix factorization and collaborative filtering techniques as possible ways of missing value estimation. We show that, given the peculiarity of VAA data, even unsuccessful for recommender systems techniques such as SVD perform quite well in estimating missing values in VAAs. We also provide online access, to the research community of VAAs and recommender systems, to two VAA datasets obtained from Cypriot presidential elections of February, 2013 and Greek parliamentary elections of May, 2012. To the best of our knowledge this is the first time the missing value problem in VAAs is approached in this, systematic, way.

2   Background

Missing value estimation is an important problem in several research areas. Whenever the cost of acquiring data or repeating an experiment is high estimating the values of missing data is the method of preference [20]. Expectation maximization (EM) and Maximum Likelihood (ML) are two well-known methods of estimating the parameters of the missing data based on the available ones. ML tries to find the underlying probability distribution of the available data. Once this is done obtaining the values of missing data is straightforward [6]. In EM the logic is similar. However, due to sparseness of available data the method estimates missing values in an iterative way. In the first step the aim is to estimate the parameters of interest from the available data and the probable values for missing data. In continue, the parameters are recalculated using the available data and the estimated values. The new parameters are applied to recalculate the missing values. The process is repeated until the values estimated from one cycle have a high correlation with those of the next [2].

EM and ML are more appropriate for unimodal probability distributions of sample. This is rarely the case in data collected from surveys and especially from VAAs as well as in recommender systems data. In those cases mixture of Gaussians are more appropriate probability distribution methods. In recommender systems this is the basic reason clustering based methods became so popular [19]. Furthermore, both ML and EM are time consuming procedures and they are used in cases where it is critical to make the best estimation irrespectively of the time required to do so. In cases where the response time is more critical than the accuracy of estimation more simple approaches are usually adopted. Single, multiple and hot deck imputation are such methods. The simplest and most common strategy for this method is calculating the value of a missing data, as being the mean for that variable. In multiple imputation approaches, each missing value is replaced by a set of imputed values. The technique which is used to generate these estimates is a simulation-based approach called Markov Chain Monte Carlo (MCMC) estimation [7]. Hot deck imputation replaces a missing value with an observed response from a similar unit [1]. The process includes finding other records in the dataset that are similar in other parts of their responses to the record with the missing values. Usually, there are a lot of data that correlate with those containing missing values. In these cases, the value that will fill the empty cell is selected randomly from the good matches and it is called donor record.

3   Problem Formulation

In this article we formulate the VAA missing value estimation as a recommender system problem. Recommender Systems (RS) are software tools and techniques that make recommendations for items that can be exploited by a user. These systems are particularly useful, since the sheer volume of data, which many modern online applications manage, often hinder users to distinguish what information is

related to their interests. So the RSs with the help of special algorithms attempt to predict what products or services a user would find interesting to see or buy.

Formally, the recommendation problem can be formulated as follows: Let C be the set of users (customers) and let S be the set of all possible items that the users can recommend, such as books, movies, restaurants, etc. Let also u be a utility function that measures the usefulness (as may expressed by user ratings) of item s to user c, i.e., $u : C \times S \to \Re$. The usefulness of all items to all users can be expressed as a matrix U with rows corresponding to users and columns corresponding to items. An entry $u(c, s)$ of this matrix may have either positive value indicating the usefulness (rating) of item s to user c or a zero value indicating that the usefulness $u(c, s)$ has not been evaluated. It should be noted, here, that although there are several cases where the rating scale is different than the one mentioned above and includes negative values (and as a result the non-evaluated items cannot be represented by the zero value) it is always possible to transform the rating scale in an interval $[l_v \ h_v]$ where both l_v and h_v are greater than zero. The recommendation problem can be seen as the estimation of zero values of matrix $U \in \Re^{N_C x N_S}$ from the non-zero ones. The quantities N_C and N_S represent the total number of users and items respectively.

Recommender systems are broadly divided into six main categories: (a) Content-based, (b) Collaborative Filtering, (c) Knowledge-based, (d) Community-based, (e) Demographic and Hybrid Recommender Systems [10]. Collaborative Filtering (CF), which is the most common RS type, assumes that if two users evaluate the same items in a similar way they are likely to have the same 'taste' and, therefore, RSs can make recommendations between them.

Recommendation in the CF approach requires some similarity $sim(c, c')$ between users c and c' to be computed based on the items that both of them evaluated with respect to their usefulness. The most popular approaches for user similarity computation are Pearson correlation and Cosine-based metrics. Both of these methods produce values $sim(c, c') \in [-1 \ 1]$. By computing the similarity of all users in pairs we create the similarity matrix $M \in \Re^{N_C x N_C}$. Zero values of matrix M may correspond to either zero similarity, or, to users with no commonly evaluated items. The influence of a user can be computed by taking the sum across the corresponding row or column of matrix M. The higher this sum is the more influential the user is.

As may one easily understand the zero values of the utility matrix U can be seen as missing values. Furthermore, in VAAs N_C can be seen as the number of users who filled in the online questionnaire while N_S is the number of questions in the questionnaire. In such a setting techniques from the recommender systems can be applied for the estimation of the missing answers.

In commercial recommender systems the utility matrix U is very sparse, that is the non-zero elements are much less than the zero ones [18]. As a result traditional approaches such as nearest neighbor and clustering based recommendation are not so effective. This is because in sparse datasets the 'neighbors' of a user may not actually coincide in preferences with him/her. Therefore, recommendations given by those neighbors are unlikely to be successful. In an effort to tackle

this problem alternative collaborative filtering approaches were proposed as well as a new category of methods based on matrix factorization. In the latter methods matrix factorization is applied to the utility matrix U in a effort to identify hidden clusters and get advantage of non-sparse areas in U. Among the matrix factorization techniques the Singular Value Decomposition (SVD) is the most widely known and it has been applied in several fields of study and especially in dimensionality reduction. Unfortunately, in sparse matrices the SVD is not so effective and as a consequence alternative iterative techniques including Alternative Least Squares (ALS) [13] and Stochastic Gradient Descent (SGD) [23] were proposed.

In VAAs the matrix U is not sparse and traditional collaborative filtering techniques as well as SVD are expected to be effective. We investigate the performance of these techniques as well as many others in an effort to estimate missing values in VAAs based on a recommender system perspective as formulated in the previous paragraphs. The results show that the majority of techniques used in recommender systems can be also, successfully, applied for missing value estimation in an elegant and mathematically strict way.

4   Approaches

In this section we briefly report the various approaches we have applied for missing value estimation in VAA data. Emphasis is given to matrix factorization and clustering-based approaches, while comparison against the simple nearest neigbor methods are also provided.

4.1   Clustering Algorithms

Clustering algorithms perform recommendation to the active user by employing a smaller set of highly similar users instead of the entire database [19]. They have been proved as an effective mean to address the well-known scalability problem that recommender system face [18]. A cluster is produced by data objects which are similar to each other. Data belonging to a cluster differ from those which belong to another cluster [8]. In the VAA setting every user is represented by a profile vector c_i which is composed of his/her answers to the online questionnaire. Given that the number of questions in the questionnaire is fixed and the possible answers come from a Likert scale the vector c_i can be encoded as a numerical one. In this way the comparison between profile vectors is straightforward by using and any similarity metric of linear algebra. Clustering of users is obtained by applying such a metric in the profile vectors.

The k-means clustering algorithm is widely used in clustering based recommendation systems mainly due to its simplicity [11]. It aims to partition N_C users, defined by the user profile vectors $\{c_1, c_2, ..., c_{N_C}\}$, into K clusters $(K << N_C)$ $\mathbf{S} = \{S_1, S_2, ..., S_K\}$. Each user is classified to a cluster according to the shortest distance between the user profile vector and the cluster's

mean vector. The user profile vectors correspond to the rows of the utility matrix U mentioned earlier. The k-means algorithm is similar to the expectation-maximization algorithm for mixtures of Gaussian in that they both attempt to find the centers of natural clusters in the data. The optimization criterion is to find the partition $\mathbf{S}^o$ that minimizes the within-cluster sum of squares (WCSS):

$$\mathbf{S}^o = \underbrace{argmin}_{\mathbf{S}}(\sum_{i=1}^{K} \sum_{c_j \in S_i} \|\boldsymbol{c}_j - \boldsymbol{\mu}_i\|^2) \tag{1}$$

where $\boldsymbol{\mu}_i$ is the mean vector of data points (user profile vectors) in cluster S_i.

By using k-means partitioning method, the missing values of a voter's answers can be estimated by the answers of the cluster members' (where the voter belongs). The facts which negatively affect that method is the sparsity of clusters, that provokes the failure of the prediction on a missing value, and the high cardinality of clusters, which increases the time required for the prediction.

4.2 Nearest Neighbor methods

In the Nearest Neighbor recommendation the most similar user of an active user is found and provides the recommendation. The main disadvantage of this method is that every time a new voter fills the online questionnaire, the similarity between voters must be recalculated. This is obviously non-scalable and time consuming. Furthermore, if the voter and his/her nearest neighbor have the same unanswered questions then prediction of missing values would be impossible. An alternative is to find the nearest neighbor among those users that have answered the question which the active user left unanswered.

The K-nearest neighbor (K-NN) algorithm is an improved extension to the Nearest Neighbor method. It limits the number of the neighbors we aim to find, by determining the value of K [4]. In case the K is not determined we can use a fixed distance alternative. In this case the K neighbors are those that fall in the hypersphere that (a) is centered on the datapoint of profile vector of the active user, and (b) has a radius equal to the fixed distance. The K nearest neighbors of the c-th voter correspond to the K voters who belong to the c-th row, of the voter similarity matrix M, with the highest values. The prediction of the voter's missing values can be calculated by using either the average of K neighbor voter answers or the median value of these answers. The most common problem with this method is that K is not always obvious since it depends on every voter separately while in the case of fixed distance K can be very small (even equal to zero).

4.3 Matrix Factorization

Matrix factorization techniques proved to be superior to clustering-based methods because they allow integration of additional information about the user. In those methods a matrix is factorized to find out two or more matrices such that when they are multiplied the result is to get back the original matrix [13].

The most common mode of matrix factorization is the calculation of a low-rank approximation to a fully controlled data matrix order to minimize the sum-squared difference of matrix values. A low-rank approximation is given from the viewpoint of the singular value decomposition (SVD) of the data matrix. The SVD expressed on a utility matrix U of size mxn leads to a factorization into three matrices: X is an mxm unitary matrix, S is an mxn rectangular diagonal matrix with nonnegative real numbers on the diagonal, and Y^* (the conjugate transpose of Y) is an nxn unitary matrix [22].

$$U = XSY^*$$ (2)

Whenever the utility matrix is large and sparse, we resort to perform Stochastic Gradient Descent (SGD) for matrix factorization. SGD has been successfully applied to large-scale and sparse machine learning problems and can handle problems with more than 10^5 training examples and more than 10^5 features [3]. SGD approximates the true gradient of $E(w, b)$ by considering a single training example at a time. We applied the algorithm for SGD as presented in [23].

Alternating Least Squares (ALS) is another technique used for matrix factorization for sparse matrices [24]. While SGD is easier and faster than ALS, ALS is more appropriate for VAA missing value estimation since the estimated values are guaranteed to real and non-negative.

5 Datasets

In our paper we have used three datasets for experimental evaluation. The first one was collected by a pre-survey for the Cypriot presidential elections 2013 conducted door to door on January 2013, the second one was collected from `www.choose4cyprus.com` and the third dataset was derived from `www.choose4greece.com`. The main characteristics of all datasets is the small number of questions and the high number of ratings per question which leads on utility matrices U of low sparsity. In all datasets the ratings are integer values in the range [1 5] corresponding to the answers 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree' respectively. We have set as missing values the instances where voters answered 'I do not have opinion' or not answered, the particular question at all. The main characteristics of these datasets are shown in Table 1. The corresponding datasets can be accessed via the URL: `www.preferencematcher.com/datasets`

6 Experimental Results and Discussion

Experiments were designed to investigate the performance of the recommendation methods, which have been reported previously, based on the accuracy of prediction of missing values. This measure computes the accuracy of predicting the values of utility matrix U using a variation of Mean Absolute Error (MAE) [9] [15] computed with the aid of Frobenius norm.

Table 1. Dataset Characteristics

	Pre-survey	Choose4Cyprus	Choose4Greece
# voters	815	18,461	75,294
# questions	35	30	30
# ratings	26,419	533,542	2,204,306
# ratings per question (average)	755	17,785	73,477
# sparsity	0.0736	0.0366	0.0247

Let $\hat{U}$ be the estimation of utility matrix U, then the accuracy measure A is defined as follows:

$$A = \frac{||U - \hat{U}||}{||U|| + ||\hat{U}||} \tag{3}$$

where $||C||$ denotes the Frobenius norm of matrix C.

Table 2 shows the MAE value of each method which used in datasets of Pre-survey, Choose4Cyprus and Choose4Greece. It can be seen in Table 2 that matrix factorization methods show better prediction accuracy than the clustering based and the nearest neighbor methods. However, the difference between matrix factorization and clustering methods is not so apparent, because of the non-sparse datasets. The best performance achieved by the SVD method, which is something expected as the datasets are not sparse and SVD very effective in non-sparse data. In addition K-NN method achieved similar results with the k-means clustering method. This result is on agreement with the results presented in [18]. We see also that, as expected, the K-NN method achieves better results than the Nearest Neighbor method. Finally, it should be noted the difference in performance of the compared methods to data collected online (Choos4Cyprus and Choose4Greece) and door to door (Pre-survey) with the former to be better. This might happen because the sparsity in the Pre-survey is larger than in the online datasets.

7   Conclusion

In this article we dealt with the problem of missing values in VAAs by using techniques from the recommender systems. We observed that the SVD method, in contrary to commercial recommender systems, is the most effective technique. This is something expected since the datasets we have used are of low sparsity. However, the effectiveness of the other methods are also high indicating that the formulation of VAA missing value problem as recommendation system is successful.

In the near future we plan to implement the SVD approach of missing value estimation in our VAAs so as to improve recommendation effectiveness by filling in unanswered questions. We are also going to investigate the effectiveness of the compared algorithms by artificially increasing the sparsity of our datasets.

Table 2. The results of missing value estimation per method. Shown values refer to accuracy A.

Method	Predicted value obtained by:	Pre-survey	Choose4Cyprus	Choose4Greece
SVD		0.0483	0.0492	0.0420
ALS		0.0516	0.0497	0.0438
SGD		0.0615	0.0616	0.0538
Nearest Neighbor		0.0897	0.0638	0.0592
k-Means	Average	0.0580	0.0496	0.0477
	Median	0.0604	0.0504	0.0474
K-NN	Average	0.0640	0.0518	0.0491
	Median	0.0700	0.0592	0.0555
	Weighted Sum	0.0657	0.0503	0.0488
Fixed Distance	Average	0.0710	0.0670	0.0651
	Median	0.0788	0.0707	0.0685
	Weighted Sum	0.0708	0.0661	0.0640

Finally, comparison with other techniques for missing value estimation such as ML and simple imputation will be also investigated. Comparison with Expectation Maximization is not necessary since the k-Means clustering is based on the same principle and has been already done in this paper.

References

1. Andridge, R. R., Little, R. J. A.: A review of hot deck imputation for survey non-response. International Statistical Review, 78(1), pp. 40-64, (2010)
2. Bilmes, J.: A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, International Computer Science Institute. Retrieved from: `http://ssli.ee.washington.edu/people/bilmes/mypapers/em.pdf` (last access March 2013) (1998)
3. Bottou, L.: Large-Scale Machine Learning with Stochastic Gradient Descent. In Lechevallier,Y., Saporta, G.(eds) Proceedings of COMPSTAT'2010. Springer-Verlag Berlin Heidelberg, August (2010)
4. Bremner, D., Demaine, E., Erickson, J., Iacono, J., Langerman, S., Morin, P., Toussaint, G.: Output- sensitive algorithms for computing nearest-neighbor decision boundaries. Discrete and Computational Geometry, 33(4), pp. 593-604 (2005)
5. Cedroni, L., Diego, G. (Eds.): Voting Advice Applications in Europe: The State of the Art. ScriptaWeb, Napoli (2010)
6. Enders, C. K.: A primer of maximum likelihood algorithms available for use with missing data. Structural Equation Modeling, 8, pp. 128-141 (2001)
7. Gamerman, D., Lopes, H. F.: Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. Chapman and Hall/CRC, 2nd edition (2006)
8. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd edition (2011)

9. Herlocker, J. L., Konstan, J. A., Riedl, J. T.: An empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms,. Information Retrieval, 5(4), pp. 287–310 (2002) .
10. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press (2010)
11. Kim, K, Ahn, H.: A recommender system using GA K-means clustering in an on-line shopping market. Expert Systems with Applications: An International Journal, 34(2), pp. 1200–1209, ACM Press, New York (2008)
12. 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)
13. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: Advances in Neural Information Processing Systems (NIPS'07), ACM Press, pp. 1257-1264 (2008)
14. Sarwar, B. M., Karypis, G., Konstan, J., Riedl, J.: Application of Dimensionality Reduction in Recommender System - A Case Study. In: Workshop on Web Mining for e-Commerce: Challenges and Opportunities (WebKDD), ACM Press (2000)
15. Shardanand, U. , Maes, P.: Social information filtering: Algorithms for automating Word of mouth. In ACM CHI 95 Conference on Human Factors in Computing Systems, pp. 210-217, ACM Press (1995)
16. Tansey, S., D., Jackson, N.: Poltics: the basics. Routledge, 4th edition (2008)
17. Triga, V., Serdult, U., Chadjipadelis, T.: Voting Advice Applications and State of the Art: Theory, Practice, and Comparative Insights. International Journal of Electronic Governance, 5(3/4) (2012)
18. 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 Articial Intelligence Tools, vol. 21, no. 4 , pp. 19-26 .World Scientific Publishing Company (2012)
19. Ungar, L., H., Foster, D. P.: Clustering Methods for Collaborative Filtering. In: AAAI Workshop on Recommendation Systems, pp. 1-16. AAAI Press (1998)
20. Vogiatzis, D., Tsapatsoulis, N.: Missing Value Estimation for DNA Microarrays with Mutliresolution Schemes. In Kollias, S., Stafylopatis, A., Duch, W., Oja, E. (eds.) Artificial Neural Networks - ICANN 2006. LNCS, vol. 4132, pp. 141-150. Springer Berlin Heidelberg (2006)
21. Walgrave, S., Van Aelst, P., Nuytemans, M.: Do the Vote Test: the Electoral Effects of a Popular Vote Advice Application at the 2004 Belgian Elections. Acta Politica, 43(1), pp. 50-70 (2008)
22. Wall, M. E., Rechtsteiner, A., Rocha, L. M.: Singular value decomposition and principal component analysis. In Berrar, D., Dubitzky, W., Granzow, M. (eds) A Practical Approach to Microarray Data Analysis. MA: Kluwer, pp. 91-109 (2003)
23. Zhou, T., Shan, H., Banerjee, A., Sapiro, G.: Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information. In SIAM International Conference on Data Mining, pp. 403-414. SIAM / Omnipress (2012)
24. Zhou, Y.: Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R. (eds) Algorithmic Aspects in Information and Management. LNCS 5034, pp. 337-348. Springer Berlin Heidelberg (2008)