Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems

Xenia Naidenova
Military Medical Academy, Russia

Dmitry Ignatov
National Research University Higher School of Economics, Russia

Information Science
REFERENCE

Managing Director:	Lindsay Johnston
Book Production Manager:	Jennifer Romanchak
Publishing Systems Analyst:	Adrienne Freeland
Managing Editor:	Joel Gamon
Development Editor:	Hannah Abelbeck
Assistant Acquisitions Editor:	Kayla Wolfe
Typesetter:	Nicole Sparano
Cover Design:	Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Diagnostic test approaches to machine learning and commonsense reasoning systems / Xenia Naidenova and Dmitry Ignatov, editors.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-4666-1900-5 (hardcover) -- ISBN 978-1-4666-1902-9 (print & perpetual access) -- ISBN 978-1-4666-1901-2 (ebook) 1. Machine learning. 2. Data mining. 3. Pattern recognition systems. 4. Computer algorithms. I. Naidenova, Xenia, 1940- II. Ignatov, Dmitry I.
Q325.5.D53 2013
006.3'12--dc23
2012004765

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 8
Bimodal Cross-Validation Approach for Recommender Systems Diagnostics

Dmitry I. Ignatov
National Research University Higher School of Economics, Russia

Jonas Poelmans
Katholieke Universiteit Leuven, Belgium

ABSTRACT

Recommender systems are becoming an inseparable part of many modern Internet web sites and web shops. The quality of recommendations made may significantly influence the browsing experience of the user and revenues made by web site owners. Developers can choose between a variety of recommender algorithms; unfortunately no general scheme exists for evaluation of their recall and precision. In this chapter, the authors propose a method based on cross-validation for diagnosing the strengths and weaknesses of recommender algorithms. The method not only splits initial data into a training and test subsets, but also splits the attribute set into a hidden and visible part. Experiments were performed on a user-based and item-based recommender algorithm. These algorithms were applied to the MovieLens dataset, and the authors found classical user-based methods perform better in terms of recall and precision.

INTRODUCTION

A modern Internet user rather frequently faces recommender systems. Recommender systems are defined by the ACM Recommender Systems conference as “software applications that aim to support users in their decision-making while interacting with large information spaces. They recommend items of interest to users based on preferences they have expressed, either explicitly or implicitly” RecSys (2011). The paper by Adomavicius et al. (2005) presented a survey on the state of the art of recommendation algorithms and grouped them in three main categories: content-based (also referred to as item-based), collaborative (also referred to as user-based), and hybrid recommendation approaches. An example of a web site where recommender systems are frequently

DOI: 10.4018/978-1-4666-1900-5.ch008

used is an online bookshop. If a user buys book X in an online book shop she also gets recommendations in the form ‘’other customers who bought book X also bought books Y and Z’’. There are also a lot of web systems which can recommend potentially interesting web sites to a particular user; they are called social bookmarking systems (e.g. http://del.ici.ou.us). Other examples include the websites http://facebook.com/ and http://twitter.com/ and for Russian companies, the websites http://imhonet.ru/ and http://www.ozon.ru/.

Besides the Internet the most popular and non-technological way to get recommendation is still friends’ suggestions. However, if a user wants more items to buy (to watch, to read etc.) the task is getting harder, because there may be a lot of different options of the choice, her friends may not be informed about latest items in the field or just have different tastes. To cope with these difficulties she can use so-called collaborative filtering Goldberg et al. (1992). Recommender algorithms based on collaborative filtering techniques utilize a fairly simple scheme. They find users of the system who have similar to her tastes or preferences, then compose the list of items the users selected and rank these items, and as a result she gets Top-N items of the list. Herlocker et al. (2004) presented in depth research on evaluating the quality of collaborative filtering approaches. Another less evident but interesting application is recommending key phrases in web advertising systems, where firms buy advertising phrases from web search engines to show advertisement by a user’s request Ignatov et al. (2008), Ignatov et al. (2008). This approach made use of Galois operators to obtain morphological association rules.

RECOMMENDER ALGORITHMS

In this paper without loss of generality we consider only two groups of recommender techniques, which can be called the classical ones, mainly user-based and item-based approaches Badrul et al. (2000), Deshpande et al. (2004). A key notion for these techniques is similarity, which can be expressed as Jacquard measure, Pearson correlation coefficient, cosine similarity etc. Initial data are usually represented by an object-attribute matrix, where the rows describe objects (users) and the columns represent attributes (items). A particular cell of the matrix can be either 1 or 0, which stands for the fact that the item was purchased or not respectively. Also the values can be rates or marks of items, for example, film’s rates given by users.

User-Based Recommendations

User-based methods find similarity between a target user u0 and other users of the recommender system. As a result the target user has n most frequently bought items by k most similar to u_0 users (customers). Let u_0 be a target user, u_0^I be items that she evaluated, $sim(u_0, u)$ be a similarity between the target user u_0 and another user u. In this research we use Pearson correlation coefficient as a similarity measure. Define the set of nearest neighbors (neighborhood) for the target user by the formula:

$$N(u_0) = \left\{ u \mid sim(u_0, u) \leq \Theta \right\}.$$

However, it is appropriate to obtain Top-k nearest neighbors, that is Top-k defines the threshold Θ. Hence the set of nearest neighbors includes k users which have similarity with u_0 higher than a certain threshold. After ordering the users by decreasing similarity, one should select not only Top-k of them, but also check the similarity value of (k+1)-th user in the list. If this similarity value is equal to the preceding one than one should add (k+1)-th user to the neighborhood $N(u_0)$. One should repeat the procedure until the next similarity value changes. Since we predict the rate of an item i by a specific target user u_0 we are interesting only those users from the neighborhood who have evaluated i:

$$N(u_0 \mid i) = \left\{u \mid i \in u^I \,\&\, u \in N(u_0)\right\}.$$

Denote by r_{ui} the rate (mark) of an item i by a user u we obtain the formula for the predicting rate

$$\hat{r}_{u_0 i} = \frac{\sum_{u\in N(u_0|i)} sim(u_0, u)\times r_{ui}}{\sum_{u\in N(u_0|i)} sim(u_0, u)}.$$

Item-Based Recommendations

The idea of the item-based algorithm is similar to the described user-based method, but similarity is calculated between items. Denote by u_0 again the target user, by u_0^I the items she evaluated, by *sim(i,j)* the similarity between items *i* and *j*. Define the neighborhood for an item *i* analogously as the neighborhood for a target user by $N(i) = \{j \mid sim(i,j) \geq \Theta\}$. By doing so we have top-k nearest items to *i*, that is top-*k* defines Θ. To predict the rate for a target user u_0 one has to compare the items which u_0 evaluated with those that she didn't rate. Therefore we refine the formula for item neighborhood taking into account the target user as follows

$$N(i \mid u_0) = \{j \mid j \notin {u_0}^I, i \in {u_0}^I, j \in N(i)\}.$$

Denote by r_{ui} the rate of an item *i* by a user *u* and by doing so we get

$$\hat{r}_{u_0 i} = \frac{\sum_{j\in {u_0}^I} sim(i,j)\times r_{u_0 j}}{\sum_{j\in {u_0}^I)} sim(i,j)}.$$

Then we rank marks in decreasing order and return the first n of them as a recommendation.

The main computational advantage of this method is based on the following fact: the number of e-commerce web-site users is usually increasing over time, but new items are added not so frequently. That is why pairwise users' similarity computation while forming a new recommendation may take much time, but items' similarity can be calculated offline in advance and the obtained similarity matrix can be reused many times later.

Item-based recommendation algorithms have some shortcomings, for instance, in case of the so-called cold start problem we don't know user's history and it's impossible to make recommendations. But in case there is users' history available the performance is typically better than that of some more sophisticated algorithms.

SIMILARITY MEASURES

To define similarity between two objects or attributes different similarity measures (or even metrics) are used. Usually, such a measure has the value between 0 and 1 (for absolute similarity). Let us consider some of these measures.

Distance-Based Similarity

To calculate similarity we should find the distance between compared objects or attributes. There are some frequently used methods to calculate the distance. Each initial object is represented by a vector in the attribute space (dually a vector of objects is used for distance calculation between two attributes). Then *Euclidean distance* between two objects x and y is defined as $d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$.

Hamming distance is usually used for binary data and is defined by the formula $d(x,y) = \sum_{x_i \neq y_i} 1$. Then, the simplest way to calculate similarity is to apply the following formula $s(x,y) = \dfrac{1}{1 + d(x,y)}$ (see, e.g. Segaran, 2007).

Let us explain this similarity calculation procedure in detail for the Hamming distance metric. In this case d may only equal natural numbers and 0, and the maximal value of s is equal to 1 for *d*=0. And for the next value *d*=1 the similarity s is equal to 1/2; it's a clear drawback of the similarity calculation formula. For example, let *x* and *y* be two binary vectors, which differ only in one component, according to the previous formula they are only one half similar. This rough character of *s* values can be easily seen in Figure 1.

Correlation as Similarity

In the formula below similarity between two vectors is calculated using the well-known Pearson correlation coefficient:

$$Pearson(x,y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2 \cdot \sum_i (y_i - \overline{y})^2},$$

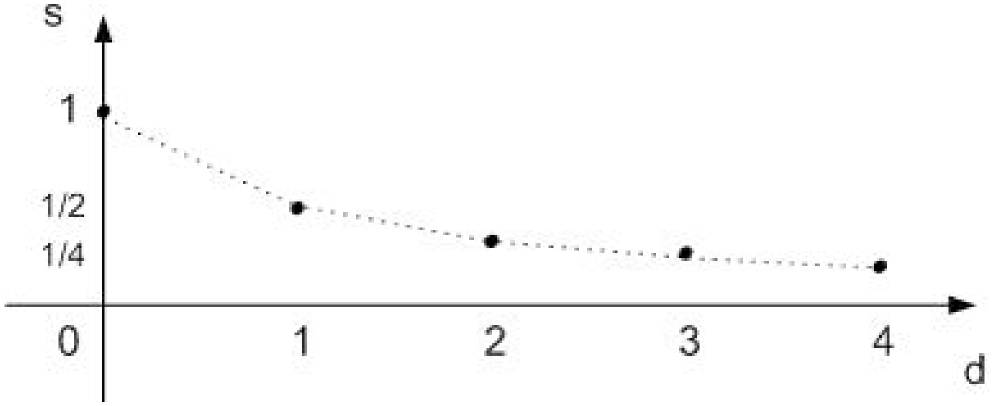
where 1 ≤*Pearson* ≤1.

The main drawback of Pearson correlation as a similarity measure is its undefined value for vectors with constant components. Moreover, we have a denominator equal to zero for the vector *x*=(4, 4, \cdots, 4). This is why we may lose some potentially relevant items for recommendation. For example, let us consider two vectors *a*=(0,5,5,4) and *b*=(0,4,5,0). If one would consider them as tuples of two users' rates then it is intuitively clear that these users are quite similar to each other. However, the correlation will not be calculated because of the following constraint: the initial vectors are trimmed to their non-zero components Symeonidis et al. (2007). In our case one should calculate the correlation between (5, 5) and (4, 5). However, as it was shown above the Pearson correlation coefficient is undefined. Some authors proposed to set the correlation value equal to 0 Segaran (2008), but in our opinion it is not correct due to possible loss of relevant items.

Other Similarity Measures

There are dozens of different measures to find the similarity, for example cosine similarity (very close to Pearson), Jacquard and Tanimoto coefficients, etc. The reader is kindly referred to Cha (2007) and Choi et al. (2010) for a complete overview.

Figure 1. Similarity versus Hamming distance



QUALITY RECOMMENDATIONS EVALUATION

In this section we propose the scheme for quality evaluation of arbitrary recommender systems. Let the initial data be represented as an object-attribute table (binary relation) $T \subseteq U \times I$, which shows that a user $u \in U$ purchased $i \in I$, i.e. uTi. To evaluate the quality of recommendations in terms of precision and recall we can split the initial user set U into training $U_{training}$ and U_{test} test subsets. The size of the test set, as a rule of thumb, should less than the size of the training set, e.g. 20% and 80% respectively. Recommendation precision and recall is evaluated on the test set. This part of the algorithm looks like one of the steps in conventional cross-validation. Then each user vector u from U_{test} is divided into two parts which consist of evaluated items $I_{visible}$ and non-evaluated items I_{hidden}. I_{hidden} are the items that we intentionally hid. Note that in the existing literature the proportion between size of $I_{visible}$ and I_{hidden} is not discussed even for similar schemes Symeonidis et al. (2007). Then, for example, a user-based algorithm can make recommendations according to similarity between users from the test and training sets. Each user from U_{test} gets the recommendations as a set of fixed size $r_n(u) = \{i_1, i_2, \cdots, i_n\}$. Precision and recall are defined by

$$recall = \frac{| r_n(u) \cap u^I \cap I_{hidden} |}{| u^I \cap I_{hidden} |},$$

$$precision = \frac{| r_n(u) \cap u^I \cap I_{hidden} |}{| r_n(u) \cap I_{hidden} |},$$

where u^I is the set of all items from I bought by the user u.

The values of these measures are calculated for each user and then averaged. The experiment is performed several times, e.g. 100, for different test and training set splits. Then the values are averaged again. In addition there is a possibility to select the I_{hidden} set, what can be done at random, but we have to specify the proportion, e.g. 20%. The idea of the method comes from machine learning where it is called cross-validation, but in case of recommender systems some modifications are necessary. Original m-fold cross validation splits the initial dataset into m disjoint subsets, where each of these subsets is used as a test set and the other subsets are considered as training ones. We modified m-fold cross-validation as described before and in addition the precision and recall computation formulas in case of division by zero. In particular, if $| u^I \cap I_{hidden} |= 0$ then recall=1. If $| r_n(u) \cap I_{hidden} |= 0$ and uI=0, then precision=1, otherwise precision=0. This approach was presented at the PerMIn 2012 and NCAI 2010 conferences (see Ignatov et al. (2010), Ignatov et al. (2012)).

Experiments have been done on the MovieLens datasets about films' rates and synthetical datasets which were generated by us.

EXPERIMENT RESULTS

We have carried out a series of experiments on the movie dataset which contains 1682 movies rated by 943 users and each of the users has evaluated at least 20 movies. All experiments were done on a laptop with Intel Core 2 Duo 2 GHz processor and 3Gb RAM with Windows Vista operating system. All algorithms were implemented in Python 2.6. We now present results of the experiment which concerns precision and recall behavior for different numbers of hidden items (10-fold cross-validation with neighborhood size 10).

As we can see from Figure 2 and Figure 3, these methods have almost identical behavior, but the recall of the user-based method is a bit higher in the range from 1 to 10 hidden attributes. In the same way, we conducted experiments on movie rates data where the percentage of hidden attributes ranged from 1% to 20%. Our diagrams show that the item-based method works better with a rather small number of hidden attributes (≈1%). For higher values of $|I_{hidden}|$ the output of the item-based method drastically decreases in quality, while the user-based method still gives quite stable results. Moreover in our experiments we observed that at 6-7% for $|I_{hidden}|$ the recall slightly increases.

We show how the quality of the results is influenced by the number of neighbors and the test set size for our synthetic data set of size 20 users × 20 items with four rectangles 5×5 full of ones (see Figure 4 and Figure 5).

We can conclude that precision and recall increase while the number of neighbors grows and the user-based method needs fewer neighbors than the item-based algorithm for achieving the same quality.

Varying the test set's size shows similar results: increasing the test set size improves the quality of prediction, and the user-based method outperforms the item-based method with respect to quality.

EXAMPLE OF REAL RECOMMENDER APPLICATIONS

Since the introduction of the so called Common State Exam in high schools of the Russian Federation, graduates received permission to apply to enter multiple universities or faculties of the same university whereas in the past they were only allowed to apply to one institution. Students are confronted with an ever increasing complexity of the educational landscape and for this purpose we developed a recommender system to guide them in their search. Students can indicate one or more faculties where they would like to study and our recommender system will make suggestions on alternative institutions in which they might also be interested. The recommender system will also use the browsing and searching history of the candidate student to efficiently suggest relevant universities, faculties, and educational directions.

A lot of techniques have been developed for recommender systems and the main principles of these algorithms are described in the previous sections. One of the most recent innovations in recommender system research is applying methods based on biclustering. In Ignatov et al. (2010a) and Ignatov et al. (2010b) a wide range of biclus-

Figure 2. Recall versus number of hidden items

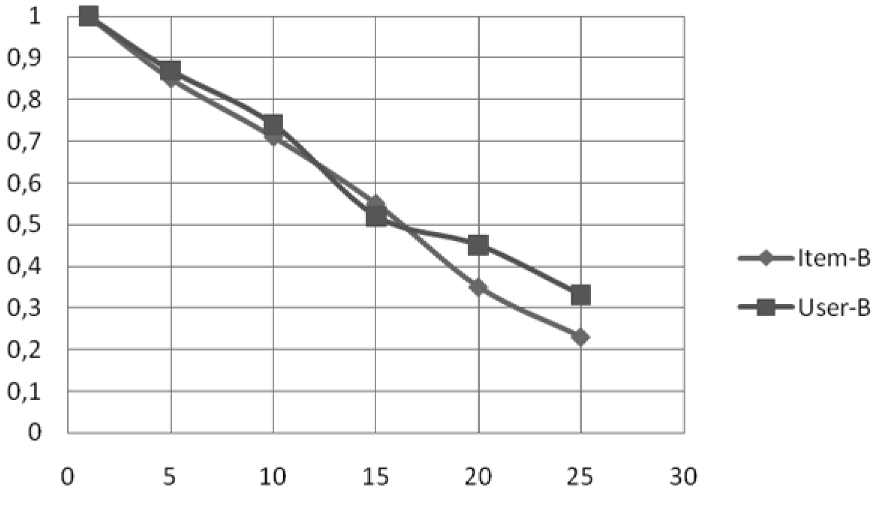
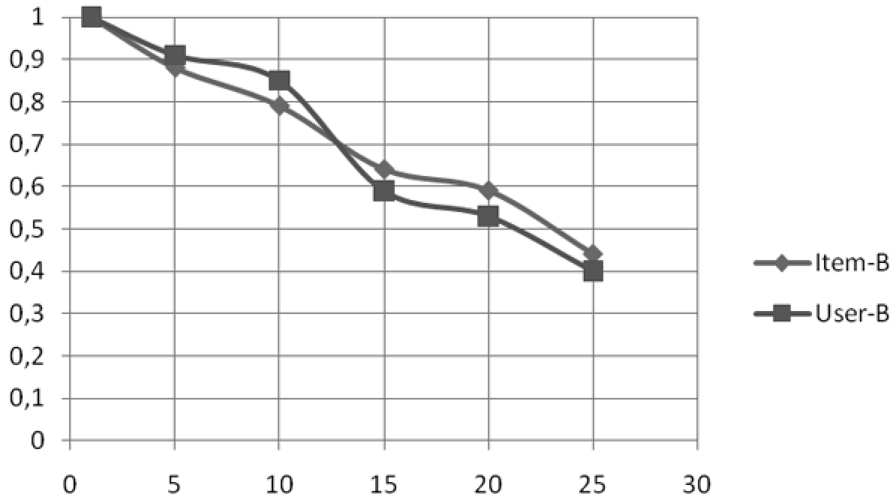


Figure 3. Recall versus number of hidden items



tering applications has been described including market research, near-duplicate web-document detection, bioinformatics etc. Biclustering is an unsupervised learning method similar to Formal Concept Analysis (FCA) Poelmans et al. (2009) and Ignatov et al. (2011). Comparing to traditional clustering methods biclustering is not a blackbox technique. Comprehensibility is one of its main advantages, i.e. it is possible to understand why objects ended up in the same cluster. For example you might ask why a cucumber and a pair of boots are assigned to the same cluster. With biclustering it can easily be revealed that they are similar because they have the same color and skin surface.

This lack of comprehensibility of traditional clustering techniques may cause serious problems in large data mining projects. To cope with these issues researchers are increasingly focusing on

Figure 4. Precision versus number of nearest neighbors

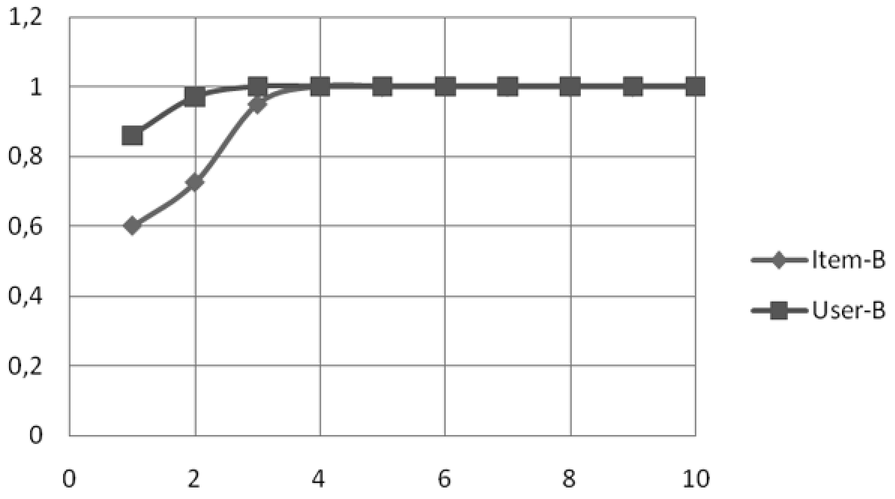
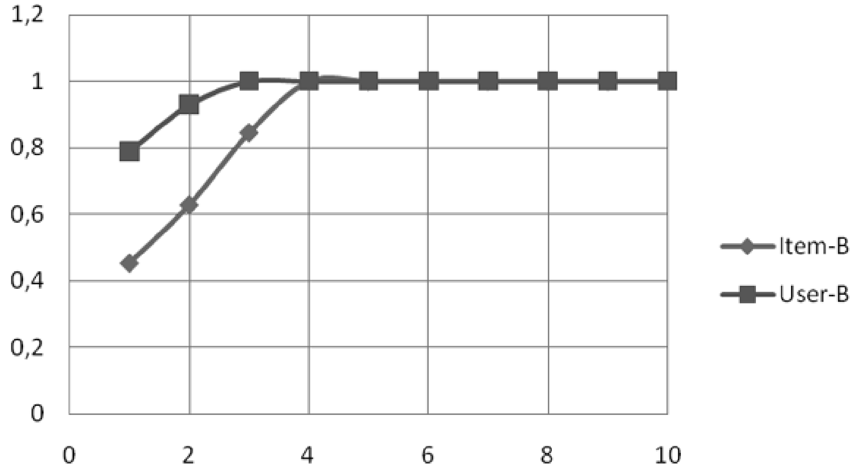


Figure 5. Precision versus number of nearest neighbors



human-centered techniques including direct clustering (John Hartigan (1972)) and Wille (1982). We chose to use biclustering instead of the more famous technique FCA because of the scalability issues encountered with FCA. The reader can find the full algorithmic description in Ignatov et al. (2011).

FUTURE RESEARCH DIRECTIONS

In the near future we plan to compare several non-traditional recently introduced recommender algorithms. More in particular these algorithms are based on biclustering, formal concept analysis and morphological association rules. Another interesting avenue is the estimation of optimal parameters for performing the bimodal cross validation approach presented in this paper. It may also be useful to investigate statistical and combinatorial properties of our approach, taking into account the execution time.

We have applications in mind for several real-life case studies. The first application will be matching of curriculum vitaes of unemployed Flemish citizens with job vacancies. In the second stage we will develop an FCA-based recommender system in cooperation with Amsterdam-Amstelland police for identifying similar incidents to a selected case (see for example Poelmans et al. (2011), Poelmans et al. (2010).

CONCLUSION

Our proposed method for evaluation of recommender algorithms makes it possible to compare the quality of the output and tune the parameter settings. We applied a user-based and item-based recommendation algorithm on the MovieLens dataset. In the experimentation on this real world data set we have found that classical user-based methods are better than item-based methods in terms of recall and precision for 10 hidden items (Top-10 is one of the most typical sizes of a recommender list). Our approach can be used for the comparison of any other recommender algorithms, e.g. biclustering based algorithms Ignatov (2008), Ignatov (2008rus).

ACKNOWLEDGMENT

Jonas Poelmans is an aspirant of the Fonds voor Wetenschappelijk Onderzoek Vlaanderen or Research Foundation Flanders. The project was partially funded by Russian Foundation for Basic Research grant no. 08-07-92497-NTSNIL_a.

REFERENCES

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

Cha, S. (2007). Comprehensive survey on distance/similarity measures between probability density. *International Journal of Mathematical Models and Methods in Applied Sciences*, *1*(4), 300–307.

Choi, S., Cha, S., & Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics Cybernetics and Informatics*, *8*(1), 43–48.

del Olmo, F. H., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Journal of Expert Systems with Applications*, *35*, 790–804.

Deshpande, M., & Karypis, G. (2004). Item-based top-*n* recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), 143–177.

Goldberg, D., Nichols, D. A., Oki, B. M., & Terry, D. B. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61–70.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, *67*(337), 123–129.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5–53.

Ignatov, D. I., Kaminskaya, A. Y., & Magizov, R. A. (2010). Cross validation for evaluation of web recommender systems. In *12th National Conference on Artificial Intelligence*, Vol. 1 (pp. 183-191). Moscow, Russia: Fizmatlit.

Ignatov, D. I., Kaminskaya, A. Y., Magizov, R. A., & Kuznetsov, S. O. (2010b). Biclustering technique based on abject and attrinute closures. In *8th International Conference on Intellectual Information Processing*, (pp. 140– 143). Cyprus: MAKS Press.

Ignatov, D. I., & Kuznetsov, S. O. (2008). Concept-based recommendations for internet advertisement. In *Sixth International Conference on Concept Lattices and their Applications*, (pp. 157–166). Palacky University, Olomouc.

Ignatov, D. I., & Kuznetsov, S. O. (2008a). Data mining techniques for Internet advertisement recommender system. In *Proceedings of 11th National Conference on Artificial Intelligence*, Vol. 2 (pp. 34 – 42). Lenand Moscow.

Ignatov, D. I., & Kuznetsov, S. O. (2010a). Biclustering of object-attribute data based on closed itemsets. In *Proceedings of 11th National Conference on Artificial Intelligence*, Vol. 1 (pp. 175-182). Moscow, Russia: Fizmalit.

Ignatov, D. I., Poelmans, J., Dedene, G., & Viaene, S. (2012). New cross-validation technique to evaluate quality of recommender systems. In Kundu, M. K. (Eds.), *PerMIn 2012, LNCS 7143* (pp. 195–202). Berlin, Germany: Springer-Verlag.

Ignatov, D. I., Poelmans, J., & Zaharchuk, V. (2011). Recommender system based on algorithm of bicluster analysis RecBi. In D. Ignatov, et al. (Eds.), *Proceedings of the International Workshop on Concept Discovery in Unstructured Data* (pp. 122-126). ISSN 1613-0073

Poelmans, J., Elzinga, P., Viaene, S., & Dedene, G. (2009). A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence. *Lecture Notes in Computer Science, 5633: Advances in Data Mining. Applications and Theoretical Aspects, 9th Industrial Conference (ICDM)* (pp. 247 – 260). Leipzig, Germany: Springer.

Poelmans, J., Elzinga, P., Viaene, S., & Dedene, G. (2010). Curbing domestic violence: instantiating C-K theory with formal concept analysis and emergent self-organizing maps. *International Systems in Accounting*. *Financial Management*, *17*(3-4), 167–191.

Poelmans, J., Elzinga, P., Viaene, S., & Dedene, G. (2011). Formally analyzing the concepts of domestic violence. *Expert Systems with Applications*, *38*(4), 3116–3130.

RecSys. (2011). Retrieved October 13, 2011, from http://recsys.acm.org/2011

Sarwar, B. M., Karypis, J., Konstan, J. A., & Riedl, J. (2001). Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce* (pp. 158 – 167).

Segaran, T. (2007). *Programming collective intelligence*. Sebastopol, CA: O'Reilly Media.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., & Manolopoulos, Y. (2007). Nearest-biclusters collaborative filtering based on constant and coherent values. *Journal of Information Retrieval*, *11*, 51–75.

Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. In Rival, I. (Ed.), *Ordered sets* (pp. 445–470). Dordrecht, The Netherlands: Reidel.

ADDITIONAL READING

Bell, R., & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. *Proceedings IEEE International Conference Data Mining (ICDM 07)*, (pp. 43-52). IEEE CS Press.

Bennet, J., & Lanning, S. (2007). *The Netflix prize, KDD Cup and workshop*. Retrieved from www.netflixprize.com

Funk, S. (2006). *Netflix update: Try this at home*. Retrieved December 2006 from http://sifter.org/~simon/journal/20061211.html

Goldberg, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*, 61–70.

Hu, Y. F., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *Proceedings IEEE International Conference on Data Mining* (ICDM 08), (pp. 263-272). IEEE CS Press.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceedings 14th ACM SIGKDD International Conference Knowledge Discovery and Data Mining* (pp. 426-434). ACM Press.

Koren, Y. (2009). Collaborative filtering with temporal dynamics. *Proceedings 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 09) (pp. 447-455). ACM Press.

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM transactions on computational biology and bioinformatics*, *1*(1), 24–45.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. *Proceedings KDD Cup and Workshop*, (pp. 39-42). ACM Press.

Poelmans, J., Elzinga, P., Viaene, S., Van Hulle, M., & Dedene, G. (2011). Text mining with emergent self organizing maps and multi-dimensional scaling: A comparative study on domestic violence. *Applied Soft Computing*, *11*(4), 3870–3876.

Salakhutdinov, R., & Mnih, A. (2008). Probabilistic matrix factorization. *Proceedings Advances in Neural Information Processing Systems* 20 (NIPS 07) (pp. 1257-1264). ACM Press.

Sarwar, B. M., et al. (2000). Application of dimensionality reduction in recommender system – A case study. *Proceedings KDD Workshop on Web Mining for e-Commerce: Challenges and Opportunities* (WebKDD). ACM Press.

Takács, G. (2007). Major components of the gravity recommendation system. *SIGKDD Explorations*, *9*, 80–84.

Zhou, Y., et al. (2008). Large-scale parallel collaborative filtering for the Netflix Prize. In *Proceedings 4th International Conference Algorithmic Aspects in Information and Management, LNCS 5034* (pp. 337-348).

KEY TERMS AND DEFINITIONS

Cross-Validation: Well-known procedure in Machine Learning for evaluation of the quality of classification algorithms.

F-Measure: The harmonic mean of precision and recall, which is a balanced value of precision and recall.

Precision: The fraction of retrieved items which are relevant.

Recall: The fraction of relevant items which are retrieved.

Recommender Algorithm: Recommender Systems are typically based on a user-based or item-based algorithm (or a more complex variant), which performs the main recommender procedure.

Recommender Systems: Software system that takes into account user preferences and other features to recommend potentially interesting items.

Similarity Measure: Mathematical function which calculates the similarity of users or items and takes values in the interval from 0 to 1.