UDC 519.68; 681.513.7; 612.8.001.57; 007.51/.52

Olena Gavrilenko,
(NTUU «KPI», Kyiv, Ukraine)

The process of computer modeling of the recommendation system for news content

The work is dedicated to the development of a common approach to the methods of data mining to provide personal recommendations, including news content. As proposed approaches to the problem are considered algorithms TF-IDF, RF, LDA, matrix factorization using SVD methods which are combined in the hybrid algorithm.

The objects of research are the electronic media news content, users, and the relationship between them.

The purpose of this work is to improve the accuracy of providing personal recommendations through the appropriate use of existing data mining methods, modified methods of building a user profile and item profile, and using hybrid recommender algorithm.

Providing personal recommendations that the problem of providing materials that are relevant to the user is an important issue in the rapid development of information technology and electronic mass media. Because millions of content characters are formed every day, the human does not have the physical capability to handle all the information. Because of the lack of a method of providing personal recommendations, valuable time is waste on searching for information, and opportunities to obtain sufficient quality information are limited. Providing personal recommendations is due to solve this problem. Recommendation systems development is versatile effort which includes experts from various fields, including data mining that is a powerful approach for the development of recommendations.

Objectives setting. Recommendation system, which consists of user elements and ratings is considered. A plurality of users of the system is denoted by *U*, and a plurality of elements – by *I*, a plurality of system retings – by *R*. In addition, *S* denotes the set of possible values for the rating (such as $S = \{1,5\}$ or $S = \{good, bad\}$. We assume that no more than one meaning of the rating could be done by any user $u \in U$ for a specific item $i \in I$, and it is denoted as r_{uv}. A subset of users, who rated the item *i*, is denoted as U_i. Similarly, I_u – subset of elements, rated by the user *u*. Finally, items that were rated by both users *u* and *v* are defined as $I_u \cap I_v$ and denoted – I_{uv}. In the same way U_{ij} denotes the set of users that rated items *i* and *j*. The most important problems of recommendation systems are the problems of the best element and *N*-the best recommendations. The first problem is to find a new element $i \in I \setminus I_u$, that is most likely to be interested for a specific user *u*. If the estimates are known, this task is often defined as a problem of regression or multi-class classification, aimed at finding the function $f : U \times I \to S$, which provides rating *f(u,i)* user *u* new element *i*.

Thus, this class of problems can be represented in terms of data mining and apply appropriate methods to solve it.The aim of this problem solving is:

– develop models of elements assessments and accuracy assessment criteriasation;
– develop and implement methods of the text preprocessing;
– develop and implement methods of data classification;
– develop and implement methods of forming personal recommendations, etc.

Methods for solving problems. Generally recommendation systems use techniques and methodologies from other neighboring areas – such as human-computer interaction or information retrieval. Nevertheless, most of these systems are basically an algorithm that can be understood as a specific example of data mining technology. Data mining process usually consists of three series of executable steps: data pre-processing, data analysis, interpretation of results (drawing 1).

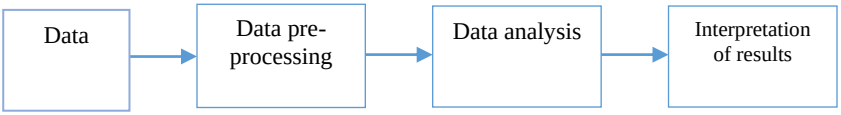

Fig.1 – Main steps of data mining

Pre-processing. Data pre-processing consists of user profile pre-processing and text content pre-processing. For effective common filtering the user needs to have sufficient content rating. We use demographic approach that can automatically detect clusters of users with common interests. Recommendations are formed from the same category of users of the same age, gender, location, interests and more. Clustering methods such as k-means method are used for creating demographic categories. After receiving the cluster for each new user, we improve the recommendations during a cold start via group guidelines and filter bots. Group method of recommendations for individual user allows to generate recommendations that most users of its demographic category rated. Group rating is calculated as follows: $GR = \prod (r_i)^{w_i}$ where r_i – rating of i user, w_i – weight of i user. The product is over all users. We provide more value to the scales w_i for users with the same data, and less value to others according to fragmentation. Filter-bots are necessary if certain data about the user is missing. This approach automatically generates some initial rankings based on available demographic data. Group recommendations will be initial ratings for each user.

Text pre-processing consists of useful content selection, rejection of stop-words and lemmatizatiDon. The first step of text pre-processing is creating of the dictionary of all the different words W, that occur in a set of the documents D, and statistics as frequent word is in each document. The order of words in a document is not considered, nor considered the context of each word. The next step is calculation of scales TF-IDF: $tfidf(w,d,D) = tf(w,d) \times idf(w,D)$, $tf(w,d) = \frac{n_{wd}}{n_d}$ where n_{wd} – the number of word occurrences w in the document d, n_d, – the total number of words in the document d, $idf(w,D) = \log \frac{|D|}{|(d \supset w)|}$, $|D|$ – the number of texts in the set, $|(d \supset w)|$ – the number of

texts where the word is found *w*. The scales TF-IDF can solve some important problems. Firstly, for too long texts only words with a maximum TF-IDF are selected, while others are discarded, thereby reducing data volumes. Secondly, the scales will be used for relevance feedback algorithm.

Algorithm RF designed to build recommendations based on text content and user ratings (excluding content evaluation). This algorithm is used only to solve the problem of cold start. The distance between the user and the document is calculated as the dot product of vectors words scales of the user and the document: $k(u,d) = \sum_{w \in W_u} V_{uv} \times tfidf(d,w)$ where W_u – the user profile words, $V_{uw} = \sum_{d \in D_u} y_{ud} \times tfidf(w,d)$, D_u – a set of the documents that have been evaluated by the user *u*, $y_{ud} \in \{-1;+1\}$ – the user rating of the document *d*. Algorithm allows to create a profile, which should be easy to interpret and can be used outside the recommendation system.

News content categorizing via Dirichlet latent allocation is an important element too [2]. The main idea LDA is that the documents are the combinations of hidden topics distributions, where each topic is defined by probability distribution on the set of words.

Model LDA allows to create a probabilistic model of a large set of data and identify hidden relationships between words by means of themes. The basis of this algorithm is naive Bayesian classifier. However naive Bayesian classifier is not enough. It is obvious that a single document can have multiple themes, but approaches which cluster document according themes do not consider this factor. Actually LDA is three-level hierarchical Bayesian network, which generates a document from a combination of themes.

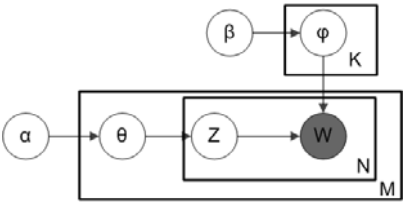

Fig. 2 – LDA model

The process is as follows. The first step is chosen for each document *d* vector of random Dirichlet distribution θ_d with parameter with parameter α. The second step is chosen theme z_{dj} with multinomial distribution with parameter θ_d. Finally, according to the chosen theme z_{dj} the word w_{dj} is chosen by distribution $\varphi_{z_{dj}}$, which is a Dirichlet distribution with parameter β, (value increase β leads to more rarefied subjects). Thus generating word model *w* from the document *d* looks like:

$p(w|d,\theta,\varphi) = \sum_{z=1}^{K} p(w|z,\varphi_z)\, p(z|d,\theta_d)$. To estimate LDA parameters Gibbs sampling is used.

Basic predictor. Common filtering model uses data on user interaction and elements to provide estimates. But most of the observed estimates specifically refers to user or element , regardless of their interaction. There are systematic trends such as providing higher ratings by user, than other users provided. So it is necessary to encapsulate those effects that are not related to user interaction with the element in the base predictor (offset). We have got estimates *m* of the users and *n* elements (components). We define indexes in order to distinguish you from the elements: for the users *u,v* , and for items *i, j ,l*. Assessment r_{ui} shows user preference to the item *u* and the higher the value, the greater the advantage. Predicted estimate value is denoted as $\hat{r}_{ui}$. A scalar value t_{ui} denotes rankings time r_{ui}. A pair *(u,i),* for which values are known refers to a plurality $K = \{(u,i)|r_{ui} \text{ are known}\}$. Each user is associated with a plurality of elements, which is denoted as *R(u),* and contains all the items that are known user rating *u*. *R(i)* is similar to the set of users who rated the item *i*. It is also possible to use a plurality, which is denoted as *N(u),* that contains all elements for which the user *u*, has an implicit preference (purchased, looked, hired etc.).

μ is denoted the average rating. Baseline predictor for unknown rating r_{ui} is denotes as $b_{ui} = \mu + b_u + b_i$ where b_u and b_i show deflection element *i* of the user *u* and the average value respectively. To estimate the value b_u and b_i the problem of the least squares is solved $\min_{b^*} \sum_{(u,i)\in K} (r_{ui} - \mu + b_u + b_i)^2 + \lambda_1 (\sum_u b_u^2 + \sum_u b_i^2)$.

This approach is easy to form the basic predictors, but does not include setting time. Most temporal variability can be included in the baseline predictors using two time effects [3]. The first effect takes into account the fact that over time the popularity of the element can be changed. It is included by processing element displacement b_i as a function of time. The second effect is that over time the user rating may change. So basic predictor b_u is also taken as a function of time $b_{ui} = \mu + b_u(t_{ui}) + b_i(t_{ui})$. That, $b_u(*)$ and $b_i(*)$ – functions that change over time. The way of this function combinations should reflect real participation of temporal parameters.

The main feature of the temporary effects is the coverage as long term and more fleeting. In most cases, results can fluctuate on a daily basis, and change over a longer period. Time function construction depends on many factors, datasets, and on purpose.

Matrix factorization. Matrix factorization model for evaluation forecasting combines the users and elements to the hidden dimensions of space *f*, so that the user's interaction is modeled as a domestic product of space. Hidden space tries to explain the assessment elements characterizing the user and the elements according to the factors that are automatically withdrawn from the user's feedback. But temporal dynamics also influences the preferences of users, and the interaction between users and the elements [4].

We construct a model of each component similar to the user's preference $p_u(t)^T = (p_{u1}(t),..., p_{uf}(t))$. Thus we can combine all the parts and extending the model SVD++, including a variable parameter: $\hat{r}_{ui} = \mu + b_u(t_{ui}) + b_i(t_{ui}) + q_i^T(p_u(t_{ui}) + \sqrt{|R(u)|}\sum_{j \in R(u)} y_j)$. Denominations b_u, b_i and p_u remain unchanged. Learning procedure is similar to the original algorithm SVD++. Thus resulting model matrix factorization allow to consider much more user features, which leads to increased accuracy and increased interpretative component.

Hybrid algorithm. To improve the efficiency and accuracy of recommender system algorithms are combined into single algorithm [5]. In this case algorithm based on content and filtering algorithm are combined: $res = w_1 * a + w_2 * b$, where w_1 and w_2– weight of the algorithm a and b suitably. Choice of weights w_1 and w_2 is done by experiment, taking values such that $w_1+w_2=1$. To assess recommender system we use accuracy parameter calculation method of use mean average square error – RMSE: $RSME = \sqrt{\frac{\sum_{(u,i)\in T}(\hat{r}_{ui} - r_{ui)^2}}{|T|}}$ where T – the total number of test assessments. The number of algorithms can be increased, but this is not advisable due to lower productivity, increasing complexity.

Conclusion. This work deals with the use of data mining methods to provide personal recommendations, including news content. Pre-processing user profile and text content were considered and use of algorithms TF-IDF, RF and LDA was offered. Matrix factorization tools and modification are provided taking into account variable time, which is important for recommendation systems. Hybrid algorithm was formed to improve the accuracy of advising and optimal use of resources and high performance providing.

References

1. Billsus D. Learning collaborative information filters [Text]: ICML '98: Proc. of the 15th Int. Conf. on Machine Learning / D. Billsus, M. J. Pazzani / – San Francisco: Morgan Kaufmann Publishers Inc., 1998. – pp. 46–54.
2. Deerwester S. Indexing by latent semantic analysis [Text] / S. Deerwester, S. T. Dumais, G. W. Furnas, R. Harshman // Journal of the American Society for Information Science. – Colorado: Deerwester, 1990. – p. 41.
3. Paterek A. "Improving Regularized Singular Value Decomposition for Collaborative Filtering" [Text]: Proc. KDD Cup and Workshop / A. Paterek / – New York: ACM, 2007. – pp. 39-42
4. Goldberg K. Eigentaste: A constant time collaborative filtering algorithm [Text] / K. Goldberg, T. Roeder, D. Gupta, C. Perkins // Journal Information Retrieval. – Berkeley: Goldberg, 2001. – pp. 133-151.
5. Burke R. Hybrid web recommender systems [Text] / R. Burke // Berlin: Springer, 2007. – p. 408.