Recommendations of Unique Items Based on Bipartite Graphs

Kristóf Marussy*
Dpt. of Computer Science and Inf. Theory
Budapest University of Techn. and Economics
1117 Budapest, Magyar Tudósok körútja 2., Hungary
marussy@cs.bme.hu

Ladislav Peska†
Faculty of Mathematics and Physics
Charles University in Prague
Malostranske nam. 25, Prague, Czech Republic
peska@ksi.mff.cuni.cz

Krisztián Buza*‡
BioIntelligence Lab
Inst. of Genomic Medicine and Rare Disorders
Semmelweis University
1083 Budapest, Tömő u. 25–29., Hungary
buza.krisztian@med.semmelweis-univ.hu

Abstract: We consider a model of recommending items to users based on weighted bipartite graphs for domains where the availability of items is limited, i.e. only a fixed number of users can acquire a single item. As a motivating example, we consider the problem of an electronic antique book store where each book can be sold to only a single customer.

Three approaches are used for recommendation generation, including greedy top-k recommendation and its generalizations k, ℓ-recommendation and k, W-recommendation. We show that the k, W-recommendation problem is NP-complete.

The recommendation methods are subjected to data-driven and stochastic evaluation. A stochastic model is used to quantify user dissatisfaction due to desired items going out of stock. Our numerical experiments have shown that greedy recommendation is outperformed by k, ℓ-recommendation and k, W-recommendation in the antique book store recommendation problem both in terms of accuracy and (expected) user satisfaction.

The more principled selection of parameters for the recommendation algorithms and the stochastic model is scope of future work, as well as applications to other domains.

Keywords: Recommender Systems, Matrix Factorization, Bipartite Matching, b-Matching

1 Introduction

Recommender Systems are software tools that provide suggestions for items to be of use to a user [12]. They are prominent components of frequently used websites, for example, if a user watches a YouTube video, the system tries to capture the user's taste and subsequently recommends similar videos. Recommender Systems have widespread use in e-commerce, where they are used to personalize visitors' user experience by providing recommendations of items the customer will most likely buy.

*This research was performed within the framework of the grant of the Hungarian Scientific Research Fund (grant no. OTKA 111710 PD and OTKA 108947 K).
†Research is supported by the grants SVV-2014-260100, P46 and GAUK-126313.
‡Research is supported by the Bolyai János Research Scholarship of the Hungarian Academy of Sciences.

In this paper, we investigate methods of recommending *unique* items, by which we mean that each item can be sold to only a single customer, because only a single piece is available at the vendor. Methods for Recommender Systems usually assume that an item can be sold to arbitrarily many customers. Examples of items with (practically) unlimited supply include mass-produced goods and digital downloads.

We suppose that the system periodically recommends a fixed number of items to each user. Additionally, the recommendations cannot be changed until the next period even if a product is sold out during the current period. Therefore, the recommendation may contain outdated items that are no longer in stock and have little chance to ever be resupplied. We will refer to the situation in which the user attempts to buy an out-of-stock item due to an outdated recommendation a *failed purchase*.

As an example, consider a trader of antique books. The trader buys second-hand books, usually single copies, then advertises them in an e-shop. The e-shop sends recommendation newsletters to registered customers every month which contains books they are likely to buy. If a single book is recommended to multiple users, only a single customer will be able to buy it even if multiple customers are interested. After the book is sold, the content of the newsletters cannot be modified to remove the book from the recommendations:

- If no customers are interested in the book, it will not be sold.
- If only a single customer is interested, she will buy the book which results in a *successful purchase*.
- If multiple customers are interested, the first to do so will buy the book by a *successful purchase*. The rest of users will face a *failed purchase* due to the book being out of stock.

Other examples of Recommender Systems with limited availability of products include event recommendation [3], where and event may be attended by only a limited number of people; online auction sites, where each auction refers to a single object; real estate agencies, where only one person can purchase the property; and recommendations in on-line dating sites, where a given user will be the romantic partner of at most one other user.

2 Representing Recommendations by Bipartite Graphs

We model the recommendation problem by a complete bipartite graph, similar to the approaches used in e.g. blog feedback prediction [2] or interaction prediction in drug–target networks.

Let A be the set of *users* and B be the set of *products*. Consider the complete bipartite graph $G = (A, B; E), E = A \times B$ with the weight function $w : E \to \mathbb{R}_{\geq 0}$. If $a_i \in A$ is a user and $b_i \in B$ is a product, we will write ij for the edge (a_i, b_j) and w_{ij} for its weight, respectively.

We make the following assumptions with regard to weights:

Assumption 1 *The weights express the users' preferences towards items. If $w_{ij} > w_{ih}$, then the user a_i is believed to prefer item b_j over b_h.*

Assumption 2 *The weights can be compared across users: if $w_{ij} > w_{tj}$, then the user a_i is more likely to be interested in b_j than the user a_t.*

Definition 3 *A* recommendation *is a subgraph $M < G$ which contains all the users, that is, $V(M) \cap A = A$. The recommended items for a given user a_i are the products which are adjacent to a_i in M. The weight of a recommendation is the sum of its edge weights, i.e. $w(M) = \sum_{ij \in E(M)} w_{ij}$.*

We will use the notation M_i for the set of items that are adjacent to the user a_i in M, and M^j for the set of users that are adjacent to item b_j ($M^j = \emptyset$ if $b_j \notin V(M)$). That is, M_i is the set of products recommended to user a_i and M^j is the set of users to whom b_j is recommended.

We want to recommend each user no more than $k \in \mathbb{N}^+$ products. In other words, $|M_i| = \deg_M a_i \leq k$ for each user a_i. Such recommendation will be called a k-recommendation.

2.1 Generating Recommendations

We consider three methods for generating the k-recommendation graph M.

2.1.1 Greedy Recommendation

Recommender Systems usually suggest the top k most preferred items for each user. This scheme maximizes the weight of the matching $w(M)$ for a given k, therefore it best reflects the users' preferences.

However, the limited availability of products in stock is ignored. In the conventional Recommender Systems setting, an item can be recommended to any number of users, because it is available in practically unlimited quantities. However, in our setting each item can be sold to at most one user. If an item b_j is highly preferred by many users, it is possible that it will be recommended to all of them due to the greediness of the algorithm, even though only one user will be able to buy it. This can lead to many failed purchases.

2.1.2 k, ℓ-Recommendation

In order to take the limited availability of the products into account, an upper bound can be placed on the number of users to whom a product b_j is recommended. Formally, it is required that $|M^j| \leq \ell$ for each product b_j given a constant $\ell \in \mathbb{N}^+$. We call such k-recommendations k, ℓ-recommendations.

Problem 4 (maximum weight b-matching) *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}$ and let b be a function $b : V \to \mathbb{N}^+$. Finding the maximum weight subgraph $M < G$ such that for each vertex $v \in V(M)$, $\deg_M v \leq b(v)$ is called the* maximum weight b-matching *or* Degree Constrained Subgraph (DCS) *[7] problem.*

Let $b(a_i) = k$ for all $a_i \in A$ and $b(b_j) = \ell$ for all $b_j \in B$. A maximum weight b-matching on the graph of user preferences G is a maximum weight k, ℓ-recommendation in G.

Efficient algorithms for maximum weight b-matchings include the extension of the blossom algorithm [7], algebraic methods [8], cost scaling, capacity scaling and network simplex methods [4].

2.1.3 k, W-Recommendation

Another way of placing and upper bound on recommendation of the same product is to limit the sum of adjacent edge weights, i.e.

$$\sum_{a_i \in M^j} w_{ij} \leq W \quad \forall b_j \in B. \tag{1}$$

We call k-recommendations that satisfy this criterion k, W-recommendations.

Problem 5 (maximum weight k, W-recommendation) *Find a k, W-recommendation of maximum weight given the weighed bipartite preference graph $G = (A, B; E)$.*

Problem 6 (k, W-recommendation decision) *Given the weighted bipartite preference graph and a positive real number $0 < w^*$, find whether there exists a k, W-recommendation M of at least w^* in weight, i.e. $w(M) \geq w^*$.*

It is straightforward to check whether a subgraph $M < G$ is a k, W-recommendation and calculate its weight $w(M)$ in polynomial time. Therefore, Problem 6 is in NP. We will now show that Problem 6 is NP-hard by Karp reduction.

Problem 7 (bin packaging decision) *Given a list of n objects with sizes $0 < s_1, s_2, \ldots, s_n \leq 1$ to pack and a positive natural number m, decide whether there exists an m-partition $S_1 \coprod S_2 \coprod \cdots \coprod S_m$ of the set $\{1, 2, \ldots, n\}$ such that*

$$\sum_{i \in S_k} s_i \leq 1 \quad \forall k = 1, 2, \ldots, m. \tag{2}$$

Bin packaging is known to be an NP-complete problem.

Proposition 8 *Bin packaging $\prec k, W$-decision, therefore k, W-decision is NP-hard.*

PROOF: If $m \geq n$, the partition

$$S_i = \begin{cases} \{i\} & \text{if } i \leq n, \\ \emptyset & \text{otherwise} \end{cases} \tag{3}$$

satisfies (2) and solves the bin packaging problem. Thus, we can reduce bin packaging with $m \geq n$ to the trivial input (G, w, W, w^*) of k, W-recommendation, where G is the empty (bipartite) preference graph, W is arbitrary and $w^* = 0$. The k, W-recommendation decision problem will output True on such input, because the empty k, W-recommendation has a weight of $0 = w^*$.

Otherwise, we can construct an input (G, w, W, w^*). Let $G = (A, B; E)$, where $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_m\}$, $E = A \times B$. That is, there is a user for each object to pack and a product for each bin the items may be packed into. This graph has $O(n)$ vertices and $O(n^2)$ edges, because $m < n$. Let $w_{ij} = s_i$, so each object "prefers" any bin according to its size.

Partitions $S_1 \coprod S_2 \coprod \cdots \coprod S_m$ satisfying (2) are in one-to-one correspondence with $k = 1, W = 1$-recommendations M in G of weight $w^* = \sum_{i=1}^{n} s_i$, where $M^j = S_j$ for all bins b_j. The weight threshold w^* is chosen so that M must contain (exactly) one edge for each object a_i, that is, $M_i = \{b_{j(i)}\}$ for all a_i. This means the object a_i is being packed into the bin $b_{j(i)}$. The graph G and the weights w can be constructed in $O(n^2)$ time, making this a polynomial reduction scheme. □

Corollary 9 *k, W-recommendation decision is NP-complete.*

Because k, W-recommendation is NP-complete, we give a heuristic method for Recommender Systems based of k, W-recommendations instead of an exact solution. We use the following greedy algorithm for k, W-recommendation:

1. Let $G = (A, B; E)$ be the complete weighted bipartite preference graph and M an empty recommendation.
2. For all edges $ij \in E$, ordered by w_{ij} nonincreasing:

   2..1.. If $w_{ij} + \sum_{a_t \in M^j} w_{tj} \leq W$ and $|M_i| \leq k$, let $M = M \cup \{ij\}$.
3. The resulting recommendation M is a k, W-recommendation, although it is not necessarily of maximum weight.

3 Content-boosted Matrix Factorization

Currently the leading algorithms for deriving recommendations are variants of Matrix Factorization, which represents state of the art in collaborative filtering techniques. However, in domains with limited item supply and sparse user interactions, such as our bookshop dataset, collaborative filtering (CF) methods face problems such as the low number of visited objects per user and the extremely sparse user-object matrix. This results in a perpetual "cold start" problem, where too little data is available for deriving useful recommendations.

We opted for compromise solution, which is Content-boosted Matrix Factorization (CBMF). CBMF leverages both feedback of other users and attributes of objects. Thus we will not miss inter-user dependencies and the cold start problem is suppressed thanks to the inter-object similarity at the same time.

We will now briefly describe the Content-boosted Matrix Factorization algorithm. For further details consult the paper by Forbes and Zhu [6].

Matrix Factorization Given the list of users $A = \{a_1, ..., a_n\}$ and objects $B = \{b_1, ..., b_m\}$, we can form the user-object rating matrix $\mathbf{R} = [\, r_{ij} \,]_{\, n \times m}$. For a given number of latent factors f, matrix factorization techniques aim to decompose original $\mathbf{R}$ matrix into $\mathbf{U}\mathbf{O}^T$, where $\mathbf{U}$ is an $n \times f$ matrix of user latent factors (μ_i^T stands for latent factors vector for particular user a_i) and $\mathbf{O}^T$ is an $f \times m$ matrix of object latent factors (σ_j is vector of latent factors for particular object b_j).

$$\mathbf{R} \approx \mathbf{U}\mathbf{O}^T = \underbrace{\begin{bmatrix} \mu_1^T \\ \mu_2^T \\ \vdots \end{bmatrix}}_{n \times f} \times \underbrace{\begin{bmatrix} \sigma_1 & \sigma_2 & \dots \end{bmatrix}}_{f \times m}. \tag{4}$$

The unknown rating for user a_i and object b_j is predicted as $\hat{r}_{ij} = \mu_i^T \sigma_j$. Our target is to learn matrices $\mathbf{U}$ and $\mathbf{O}$ while minimizing errors on known ratings. Regularization penalty is added to prevent overfitting resulting in the optimization equation (5).

$$\min_{\mathbf{U},\mathbf{O}} \| \mathbf{R} - \mathbf{U}\mathbf{O}^T \|^2 + \lambda (\| \mathbf{U} \|^2 + \| \mathbf{O} \|^2). \tag{5}$$

Content boosted matrix factorization (CBMF) is a method based on the assumption that an object can be represented merely by the list of its attributes. The assumption gives us the advantage of facilitating attributes directly into matrix factorization, but on the other hand ignores possible hidden object features (e.g. evaluation of design, previous user experience with given object, etc). More formally let $\mathbf{T}_{m \times t}$ be a matrix of object attributes and $\mathbf{B}_{t \times f}$ a matrix of latent factors for each attribute. The attributse constraint can be then formulated as

$$\mathbf{O} = \mathbf{T}\mathbf{B}. \tag{6}$$

By substituting $\mathbf{T}\mathbf{B}$ for $\mathbf{O}$ in the rating matrix (4), we can reformulate the factorization problem as follows:

$$\mathbf{R} \approx \mathbf{U}\mathbf{O}^T = \mathbf{U}\mathbf{B}^T\mathbf{T}^T = \underbrace{\begin{bmatrix} \mu_1^T \\ \mu_2^T \\ \vdots \end{bmatrix}}_{n \times f} \times \underbrace{\mathbf{B}^T}_{f \times a} \times \underbrace{\begin{bmatrix} t_1 & t_2 & \dots \end{bmatrix}}_{t \times m}. \tag{7}$$

The minimization equation (5) can be reformulated using Stochastic Gradient Descent, which moves along the gradient of the objective function (5). First, the gradient with respect to $\mathbf{U}$ is calculated while $\mathbf{B}$ is fixed and we move along the gradient towards the optimum according to the learning rate η. Next, the gradient according to $\mathbf{B}$ is used for optimization. The two steps are alternated until convergence.

The iterative formulas for CBMF are

$$\begin{aligned} \mu_i &= \mu_i + \eta \left(\sum_{(i,j) \in K} (r_{ij} - \mu_i^T \mathbf{B}^T t_j) \mathbf{B}^T t_j - \lambda \mu_i \right), \\ \sigma_j &= \sigma_j + \eta \left(\sum_{(i,j) \in K} (r_{ij} - \mu_i^T \mathbf{B}^T t_j) t_j \mu_i^T - \lambda \mathbf{B} \right), \end{aligned} \tag{8}$$

where K is the set of all observed user–object interaction pairs.

4 Evaluation

4.1 Data Set

The evaluation procedure was conducted on a dataset from a Czech second-hand bookshop called Antikvariat Ichtys*. According to the user traffic and purchase rate, the bookshop belongs to the smaller e-commerce enterprises. It offers approximately 10000 objects with following attributes: *book title*, *author name*, *book category*, *publisher*, *publishing date*, *short textual description* and *book price*. Most of the books are available only in a single copy. The bookshop attracts approximately 50–100 unique users daily and 0–3 books are bought. The dataset based on the bookshop was populated during the period of February 2014 to November 2014. The dataset contains in total approximately 22000 records of interactions between users and objects, 17000 unique users, 6300 unique objects and 150 purchases.

4.2 Experimental Protocol

In our numerical experiments, we split the interactions —i.e. item page visits and purchases— performed by each user in half according to the time of the activity. The earlier half of the interactions are added to the *training set*, while the later half are added to the *evaluation set*. We discarded users with less than 5 interactions, which resulted in 2149 users being considered. The training set contains 9851 interactions concerning 3755 books, while the evaluation set contains 8579 interactions concerning 3731 books.

User preferences are extracted from the training set by Content Boosted Matrix Factorization [6] as described in Section 3.

We compare the greedy, k, ℓ- and k, W-recommendation approaches to book recommendation by *data-driven* and *stochastic evaluation*. In both tasks, the Recommender System recommends $k = 100$ items to each user from the evaluation set.

Weight Matrix Generation Due to the sparsity of the available user interaction and purchase data [10], we use Content Boosted Matrix Factorization [6] to generate the weights w_{ij} that describe user preferences.

The title, author, year of publication and category of the books were used as additional information for the Content Boosted Matrix Factorization algorithm.

The number of latent factors were $n_f = 10$ and the regularization coefficient was chosen as $\lambda = 10$. The step size for Stochastic Gradient Descent was determined by backtracking line search [1].

Data-Driven Evaluation In the data-driven evaluation of the Recommender Systems, we compare recommendation M to the set of actual page visits and purchases by users in the evaluation period. We denote the number of page visits accurately predicted by M, i.e. the presence of page visits, by v_M and number of purchases accurately predicted by s_M.

This constitutes the standard evaluation criteria of Recommender Systems. However, by looking at only the successful purchases, the user dissatisfaction due to failed purchase attempts, i.e. situation where a user follows a recommendation referring to an item no longer available, cannot be measured. Such data-driven experiment could be carried out only by logging failed purchase attempts on the e-shop website.

Stochastic Evaluation We propose a simple probabilistic model to predict user dissatisfaction based on the expected number of failed purchases. The expected value of S_M, the number of items sold *due to the recommendation* M, and the expected value of F_M, the number of failed purchase attempts due to M, is calculated and compared between recommendation methods.

The stochastic evaluation method is described in detail in the Appendix.

*www.antikvariat-ichtys.cz

		Data-Driven		Stochastic		Scaled	
		v_M	s_M	$\mathbb{E}[S_M]$	$\mathbb{E}[F_M]$	$\mathbb{E}[S'_M]$	$\mathbb{E}[F'_M]$
Greedy recommendation		392	12	9.19	0.359	86.65	55.62
k, ℓ-recommendation	$\ell = 100$	950	19	7.58	0.014	110.02	3.12
	$\ell = 300$	732	15	8.65	0.053	118.72	11.04
	$\ell = 500$	639	17	8.87	0.087	115.98	17.60
	$\ell = 700$	584	13	8.99	0.122	112.04	23.79
	$\ell = 900$	557	13	9.07	0.157	107.94	29.52
k, W-recommendation	$W = 50$	409	8	8.55	0.061	115.79	12.53
	$W = 70$	434	9	8.61	0.067	115.68	13.63
	$W = 90$	430	7	8.66	0.073	115.42	14.66
	$W = 200$	434	8	8.81	0.100	112.91	19.77
	$W = 500$	406	11	8.97	0.165	105.58	30.58
Random		190	7	5.37	0.004	79.07	1.04

Table 1: Summary of the evaluation. The performance measure v_M and s_M denote the number of accurately predicted page visits and purchases, while S_M and F_M denote the expected number of sold items and failed purchase attempts, respectively. The Scaled version of the stochastic evaluation simulates a sales volume 10 times greater than actual

4.3 Implementation

Content Boosted Matrix Factorization, greedy and k, W-recommendation generation and the evaluation tool where implemented in R [11]. We used the Aggressive Czech Stemmer [5, 13] for the stemming of book titles and the implementation of the *network simplex* algorithm from the LEMON Open Source Graph Template Library [4].

The experiments were run on a PC with a 2.00 GHz Intel® Core™ i7 processor and 8 GB memory.

4.4 Results

Table 1 summarizes the results of our numerical experiments.

Both k, ℓ-recommendations and k, W-recommendations outperformed the greedy recommender in the data-driven and stochastic evaluations. While the greedy matching had the largest number of expected purchases ($\mathbb{E}[S_M] = 9.19$), it actually predicted page visits and purchases with the lowest accuracy. Changing the ℓ and W parameter in k, ℓ- and k, W-recommendations also resulted in a decrease prediction accuracy, while the expected number of purchases increased. This suggests that our simple stochastic model failed to fully capture user behaviour.

We repeated the stochastic evaluation by setting parameters corresponding to an actual sales volume of 1940 sold books instead of 194. The results of this evaluation are summarized in the Scaled column of Table 1. This corresponds to a simulated tenfold increase in book sales. In this evaluation, the change in the expected number of books sold is more correlated with the accuracy of the recommender systems. The expected number of failed purchases due to out-of-stock items, which lead to user dissatisfaction, also shows more dramatic contrasts: approximately 55 failed purchases are expected with the greedy recommendation compared to the 3 failed purchases with k, ℓ-recommendation with $\ell = 100$.

Figure 1 summarizes the performance of $k, \ell-$ and k, W-recommendations as their parameters are changed. The k, ℓ-recommendation method was up to 2 times more accurate than the k, W-recommendation method, while it also has fewer expected failed purchases. However, this may be caused by the relatively naive heuristic we employed, and needs further investigation.

All the considered recommendation methods performed better than the random recommender, which assigns books to users arbitrarily, except in terms of expected failed purchases. This is caused by the fact that when users are recommended items they are not interested in, there will be very few attempted purchases. Therefore, there will be little contention for items.

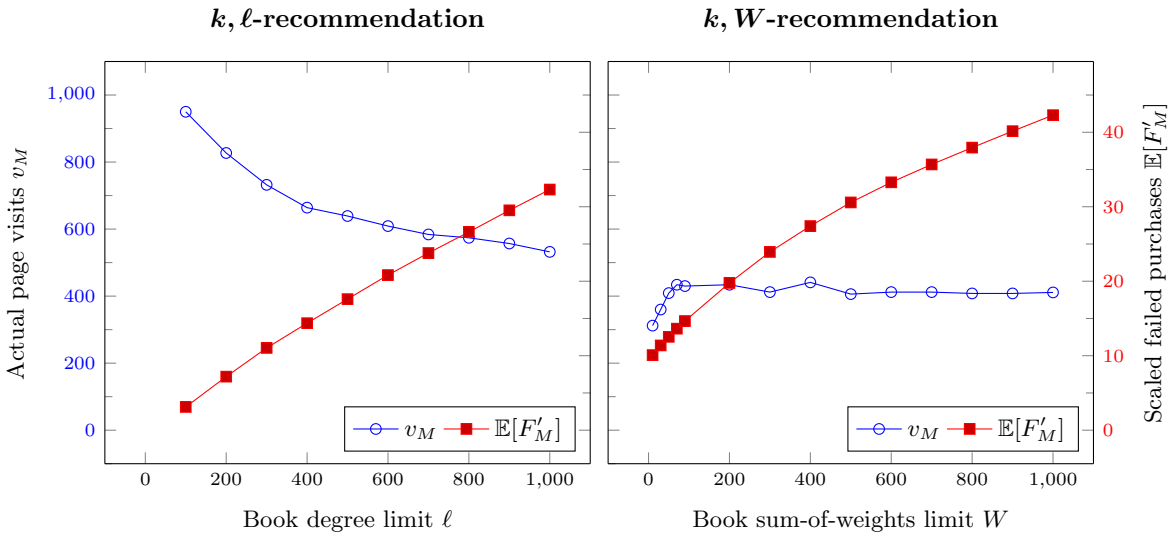

Figure 1: The number of accurately predicted page visits v_m and the expected number of failed purchases $\mathbb{E}[F'_M]$ of k, ℓ- and k, W-recommendations. Increasing ℓ decreases both accuracy and user satisfaction, while increasing W may increase accuracy at the cost of decreasing user satisfaction

5 Conclusions and Future Work

We have shown that Recommender Systems based on a novel weighted bipartite matching representation of recommendations can outperform greedy method in the antique bookshop problem, where both sparsity of user feedback and the limited availability of items pose a challenge for recommendation algorithms.

We used Content Boosted Matrix Factorization to handle the sparse feedback problem, while we compared greedy, k, ℓ- and k, W-recommendations for the limited availability problem. Our numerical experiments have shown that our matching-based methods are simultaneously more accurate than the greedy method and may provide better user satisfaction.

An interesting problem is the selection of parameters ℓ and W, for which more principled approaches shall be the scope of future work. An additional challenge is the development of an approximation algorithm based on stronger heuristics for k, W-recommendation, which could make it more competitive.

In order to quantify user satisfaction, we used a simple linear stochastic model to describe purchase attempts by users. Actual data regarding dissatisfaction and failed purchase attempts may be difficult to obtain, therefore, probabilistic models could help both parameter selection for the Recommendation Systems as well as the development of new approaches.

A Appendix

A.1 Linear Model of User Behaviour

We model user's behaviour with a simple linear model, which is consistent with Assumptions 1 and 2.

Assumption 10 *A user a_i will attempt to purchase item b_j with probability $\pi_{ij} = \min(\max(\alpha w_{ij}, 0), 1)$. The purchase will succeed if and only if the user is the first to attempt purchasing b_j, which means that only a single copy is available of every item. The purchase attempts are independent from each other.*

The parameter α is not needed to be a constant in practice, for example, people tend to buy more items during the holiday season. Moreover, Assumption 2 may not hold if the quantity of purchases differ

greatly among users; for such situations, the value of α should depend on the user. Therefore, we will regard α as unknown for the purposes of recommendation generation and will work only with w_{ij}.

A.2 Finding α

For the purpose of evaluating a recommendation system, we can approximate α as follows: Let S_j be the indicator random variable

$$S_j = \begin{cases} 1 & \text{if item } b_j \text{ is sold in the evaluation period,} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Because at least one purchase attempt is required for an item to be sold,

$$\Pr(S_j = 1) = 1 - \prod_{a_i \in A} (1 - \pi_{ij}). \tag{10}$$

Now let us consider the number of items sold in the evaluation period $S = \sum_{b_j \in B} S_j$. We will choose α such that the expected number of purchases $\mathbb{E}[S] = \sum_{b_j \in B} P(S_j = 1)$ is equal to the number of *actual* purchases s. This corresponds to a method of moments type estimation of α.

A.3 Expected Purchases and Failed Purchase Attempts

In the stochastic evaluation we calculate expected value of S_M, the number of items sold *due to the recommendation* M, and F_M, the number of failed purchase attempts due to M.

Assumption 11 *We will assume that users $a_i \in M^j$, who were recommended the item b_j, will decide whether they want to buy b_j before any user $a_t \notin M^j$. In other words, users $a_t \notin M^j$ can find the item b_j only if it was not sold to any user $a_i \in M^j$ already.*

This assumption means that only the user–item pairs that are in M need to be considered when calculating S_M and F_M, because if any user $a_i \in M^j$ attempts to purchase the item b_j, the item will be sold to some user in M^j.

The expected value $\mathbb{E}[S_M]$ can be calculated in a manner similar to $\mathbb{E}[S]$,

$$\mathbb{E}[S_M] = \sum_{b_j \in B} \left[1 - \prod_{a_i \in M^j} (1 - \pi_{ij}) \right]. \tag{11}$$

Let us define the random variable P_{ij} as the indicator

$$P_{ij} = \begin{cases} 1 & \text{if user } a_i \text{ made a purchase attempt at item } b_j, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

According to Assumption 10, the purchase attempts P_{ij} are independent Bernoulli trials with probabilities $\Pr(P_{ij} = 1) = \pi_{ij}$. Let the number of purchase attempts due to the recommendation M be denoted as $P_{Mj} = \sum_{a_i \in M^j} P_{ij}$. Because it is a sum of independent Bernoulli indicators, P_{Mj} is distributed according the Poisson binomial distribution. Its distribution may be approximated by a Poisson distribution according to Le Cam's theorem [9],

$$\Pr(P_{Mj} = v) \approx \frac{(\lambda_{Mj})^v}{v!} e^{-\lambda_{Mj}}, \qquad \lambda_{Mj} = \sum_{a_i \in M^j} \pi_{ij}. \tag{13}$$

Now we can write the expected number of failed purchases as

$$\mathbb{E}[F_M] = \sum_{b_j \in B} \sum_{v=2}^{|M^j|} (v-1) \Pr(P_{Mj} = v) \approx \sum_{b_j \in B} \sum_{v=2}^{|M^j|} (v-1) \frac{(\lambda_{Mj})^v}{v!} e^{-\lambda_{Mj}}, \tag{14}$$

that is, if v users attempt to purchase the item b_j, there will be $v - 1$ failed purchase attempts.

References

[1] L. Armijo, Minimization of functions having lipschitz continuous first partial derivatives., *Pacific J. Math.* (1966) **16**, pp. 1–3

[2] K. Buza and I. Galambos, An application of link prediction in bipartite graphs: Personalized blog feedback prediction, in *8th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications June 4-7, 2013, Veszprém, Hungary*, p. 89

[3] A. Q. de Macedo and L. B. Marinho, Event recommendation in event-based social networks, in *Hypertext 2014 Extended Proceedings: Late-breaking Results, Doctoral Consortium and Workshop Proceedings of the 25th ACM Hypertext and Social Media Conference (Hypertext 2014), Santiago, Chile, September 1-4, 2014.*, CEUR Workshop Proceedings, CEUR-WS.org (2014) **1210**

[4] B. Dezső, A. Jüttner, and P. Kovács, Lemon–an open source C++ graph template library, *Electronic Notes in Theoretical Computer Science* (2011) **264**, pp. 23–45

[5] L. Dolamic and J. Savoy, Indexing and stemming approaches for the czech language, *Inf. Process. Manage.* (2009) **45**, pp. 714–720

[6] P. Forbes and M. Zhu, Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation, in *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, ACM (2011), pp. 261–264

[7] H. N. Gabow, An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems, in *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, ACM (1983), pp. 448–456

[8] H. N. Gabow and P. Sankowski, Algebraic algorithms for b-matching, shortest undirected paths, and f-factors, *CoRR* (2013) **abs/1304.6740**

[9] L. Le Cam et al., An approximation theorem for the poisson binomial distribution, *Pacific J. Math* (1960) **10**, pp. 1181–1197

[10] L. Peska and P. Vojtás, Enhancing recommender system with linked open data, in *Flexible Query Answering Systems - 10th International Conference, FQAS 2013, Granada, Spain, September 18-20, 2013. Proceedings*, Lecture Notes in Computer Science, Springer (2013) **8132**, pp. 483–494

[11] R Core Team *R: A Language and Environment for Statistical Computing* R Foundation for Statistical Computing Vienna, Austria (2014)

[12] F. Ricci, L. Rokach, and B. Shapira, Introduction to recommender systems handbook, in *Recommender Systems Handbook*, Springer (2011), pp. 1–35

[13] J. Savoy, IR multilingual resources at UniNE (2010), http://members.unine.ch/jacques.savoy/clef/, accessed 29 January 2015