2015 IEEE 15th International Conference on Data Mining Workshops

Similarity Learning for Product Recommendation and Scoring using Multi-Channel Data

Iftikhar Ahamath Burhanuddin
Adobe Research
Bangalore, India
Email: burhanud@adobe.com

Payal Bajaj
Adobe Research
Bangalore, India
Email: pabajaj@adobe.com

Sumit Shekhar
Adobe Research
Bangalore, India
Email: sushekha@adobe.com

Dipayan Mukherjee
Indian Institute of Technology
Kharagpur, India
Email: dipayan1992@gmail.com

Ashish Raj
Indian Institute of Technology
Kanpur, India
Email: ashishrj.162@gmail.com

Aravind Sankar
Indian Institute of Technology
Madras, India
Email: aravindsankar28@gmail.com

***Abstract*—Customers may interact with a retail store through many channels. Technology now makes it is possible to track customer behavior across channels. We propose a system where items are recommended based on learning channel specific similarities between customers and items. This is done by treating recommendations as a learning to rank problem and minimizing *rank loss* with surrogate loss functions. We build our system using a real world multi-channel data set – online browse and purchase, and in-store purchase – from a retail chain. The results show that using learned similarity scores improves the performance of the system over scores generated using standard cosine similarity measures. Finally, using our learning to rank formulation we introduce a product scoring system to measure consumption behavior.**

I. Introduction

Customers may interact with a retail store through many channels — in-store, online, social media, email campaigns, etc. With the right infrastructure in place, it is now possible to track user behavior across channels. What customers purchase online for instance may prove to be rich data source to predict their in-store purchases. In this paper, we report on the design and performance of a system which uses data from three channels – online browse and purchase, and in-store purchase data – to recommend products.

Our system is a content based recommendation system which employs the pair-wise approach to the learning to rank problem. The ranking function is learned from the data by minimizing *rank loss*, which equals 1-AUC, the complement of the area under the ROC curve. The baseline model we choose to evaluate our system against is one where channel-specific scores are computed using the standard cosine similarity. In both our system and the baseline model, channel specific similarity scores induce a ranking on the set of items for each user. These rankings are aggregated by summing the scores from each of the channels. Top-k item scores determine the recommendations for a user.

We proceed to enumerate the contributions of this paper. Firstly, from an empirical stand-point, using a multi-channel marketing dataset, we demonstrate that learning similarity measures improves AUC over using the standard cosine similarity. In addition, we evaluate the performance of several loss functions in our optimization framework. Next, from a theoretical perspective, the application of similarity learning and transfer learning techniques to multi-channel data is novel. Lastly, we devise a product scoring system which measures how products are consumed in a multi-channel scenario.

In Section II we discuss literature which has inspired the ideas in this paper. This is followed by the problem definition. Section IV introduces our similarity learning optimization algorithm and Section V describes our experimental setup and showcases results. Section VI introduces our product scoring system.

II. Related Work

Similarity learning has been studied in many contexts including face verification and music recommendation as we discuss below. In [15], the authors use cosine similarity as a measure of the distance between two face vectors for the problem of face verification, where the task is to predict whether two images are from the same person. Their method learns a transformation matrix from the training data so that cosine similarity performs well in the transformed space. We employ this idea in our system. The paper [4] formulates the problem of unconstrained face verification as a similarity metric learning convex optimization problem.

McFee et al. have worked on learning to rank from a metric and similarity learning perspective [12], [13] with the latter being applied to content based music recommendation. Recently, a multiple kernel based similarity learning method for multi-modal data was described in [14].

Our work relates to the problem of rank aggregation as it creates a consensus ranking of items for each user by aggregating rankings from multiple channel via similarity scores. The reader might consider reading the paper [8] to learn about rank aggregation and its connection to the standard problem of voting.

The problem of learning from multiple sources has also been well studied from a machine learning perspective. Blum

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.139

and Mitchell [3] first proposed a framework for learning from unlabeled web data. Similarly, transfer learning addresses the problem of transferring information from a richer, labeled domain to a domain with relatively less labels. A survey of these methods can be found in [16]. Specifically, for product recommendation, algorithms based on transfer learning have been explored in [11], [18], [17].

The technique of minimizing a convex surrogate of a $0-1$ loss function to make the optimization algorithm computationally efficient is a popular one in machine learning [2]. The specific problem of optimizing AUC has gained the research community's attention following recent work on rank risk and rank regret bounds [9].

Our approach which is immensely inspired by the work of [10], where the authors approach the problem of recommendations via colloborative ranking by assuming the rating matrix is locally low-rank within certain neighborhoods of the metric space defined by (user, item) pairs. They combine an approach for local low-rank approximation based on the Frobenius norm with a general empirical risk minimization for ranking losses using several loss functions. In contrast, we investigate a content-based ranking system for recommendation, where similarities are learned from the data to minimize pairwise ranking loss. Related work on this topic includes the paper [1] by Balakrishnan et al., where the authors optimize for the nDCG metric using point-wise and pair-wise methods techniques from the learning to rank literature.

We recall that ROC curves are generally used to present accuracy for binary decision problems, however, for highly skewed datasets, Precision-Recall (PR) curves give a more informative picture of an algorithm's performance [6]. Furthermore, algorithms that optimize the area under the ROC curve (AUC) are not guaranteed to optimize the area under the PR curve. We believe it is an open problem on how our approach can be adapted to the PR curve.

III. Problem definition

We recall the notation introduced in [10] and adapt it to our context. Let the set of users and items be denoted by $\mathscr{U}$ and $\mathscr{I}$ respectively. Let each user and each item being be represented as vectors in $\mathbb{R}^d$ as is typical in the content based recommendation setting. A *similarity* function $f : \mathscr{U} \times \mathscr{I} \to \mathbb{R}$ returns a real-valued score indicating how similar a user is to an item, with a score higher in value indicating greater similarity between the user-item pair than one which is not. Restricting f to a specific user u induces a preference function on the set of items, that is, $f(u,i) > f(u,j)$ implies u prefers item i over item j. The function f is learned from the data such that it minimizes the risk function

$$\mathscr{E}(f) = \sum_{u \in \mathscr{U}} \sum_{i,j \in \mathscr{I}} \mathscr{L}(f(u,i) - f(u,j), M_{u,i} - M_{u,j}), \quad (1)$$

where $\mathscr{L}$ is a loss function and $M_{x,y}$ denotes whether user x consumed item y in one channel. We will soon extend our exposition to cover the multi-channel scenario. A popular choice for $\mathscr{L}$ is the $0-1$ loss function, which counts sign disagreements between the arguments of $\mathscr{L}$. We introduce Δ notation to denote differences of similarity function values.

$$\Delta f(x,y,z) = f(x,y) - f(x,z) \quad (2)$$

Let $\mathscr{I}_u^+$ denote the set of items consumed by u and $\mathscr{I}_u^-$ denote the complement of $\mathscr{I}_u^+$ with respect to $\mathscr{I}$. The activity of consumption will depend on the interaction channel being discussed, namely product purchases when in-store (IP) and online purchase (OP) channels are considered and product browsing behavior when online browse channel (OB) is analyzed.

To extend our discussion to a multi-channel setting, we investigate a family of risk functions. The formulation of all of these functions is driven by the idea that the item recommendations for a particular user in our system are determined by a weighted sum of the user's channel similarity scores. This makes us restrict the function f to be

$$f = w_{\mathrm{IP}} f_{\mathrm{IP}} + w_{\mathrm{OP}} f_{\mathrm{OP}} + w_{\mathrm{OB}} f_{\mathrm{OB}}, \quad (3)$$

where the weights impose a preference order on the channel similarity scores. The channel specific f_c's are similar to f in that they take a user u and an item i as arguments. The user vector u refers to the user's consumption activity in channel c. We proceed to formulate three risk functions, namely MLSI, MLST and MLSH. The abbreviation MLS stands for Multi-channel Similarity Learning and the letters I, T and H stand for Individual, Total and Hybrid. Since the objective of the risk functions is maximizing AUC in a target channel, this can be identified as a special case of transfer learning [16]. In this case, we simply force different channels to share the target label, as all the channels have the same feature space.

We refer to the sets $\mathscr{I}_u^+$ and $\mathscr{I}_u^-$ as *label* sets since they partition the set of items into consumed and not consumed items for each user. In this paper, the label sets primarily refer to consumption activity in the target channel, which is in-store purchase (IP).

The choices made so far simplify Eq. (1) to the following form for MLSI function

$$\mathscr{E}_{MLSI}(f_{\mathrm{IP}}, f_{\mathrm{OP}}, f_{\mathrm{OB}}) = \sum_{u \in \mathscr{U}} \sum_{c} \sum_{i \in \mathscr{I}_u^+} \sum_{j \in \mathscr{I}_u^-} {}_{\Delta f_c(u,i,j) < 0}, \quad (4)$$

where c is summed over the channels IP, OP, OB and we suppress channel notation from the item consumption sets.

The MLST function look like

$$\mathscr{E}_{MLST}(f_{\mathrm{IP}}, f_{\mathrm{OP}}, f_{\mathrm{OB}}) = \sum_{u \in \mathscr{U}} \sum_{i \in \mathscr{I}_u^+} \sum_{j \in \mathscr{I}_u^-} {}_{\Delta f(u,i,j) < 0}. \quad (5)$$

The MLST formulation does not consider out-of-order instances of consumed and not consumed items within each channel for each user. To address this situation, we introduce a hybrid function MLSH, which is defined as

$$\begin{aligned} \mathscr{E}_{MLSH}(f_{\mathrm{IP}}, f_{\mathrm{OP}}, f_{\mathrm{OB}}) = & \sum_{u,c,i,j} {}_{\Delta f_c(u,i,j) < 0} \\ & + \sum_{u,i,j} {}_{\Delta f(u,i,j) < 0} \end{aligned} \quad (6)$$

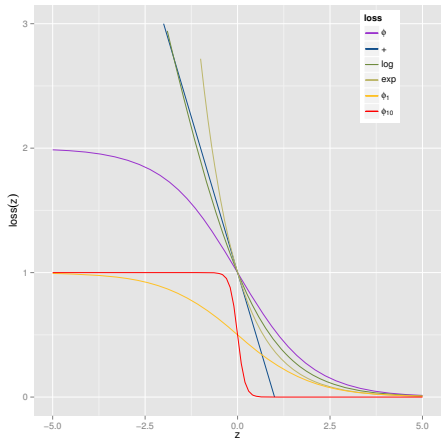


Fig. 1. Loss functions

In the above equation u, i, j are summed over the same sets as before; we abbreviate notation for brevity.

The problem we seek to solve is to find optimal ranking functions for each channel which minimizes these risk functions. The functions $\mathscr{E}$ are computationally hard to optimize for large problem instances due to their discrete nature. Instead, we replace the indicator function $_{z<0}$ with surrogate loss functions. This opens the door for numerical optimization techniques which we investigate in subsequent sections. The loss functions we consider in this paper are

- sigmoid loss

$$\ell_\phi(z) = \frac{2}{1+e^z}, \tag{7}$$

- hinge loss

$$\ell_+(z) = \max(1-z, 0), \tag{8}$$

- logistic loss

$$\ell_{\log}(z) = \log_2(1+e^{-z}), \tag{9}$$

- exponential loss

$$\ell_{\exp}(z) = e^{-z}, \tag{10}$$

- sigmoid-β loss

$$\ell_{\phi_\beta}(z) = \frac{1}{1+e^{\beta z}}, \tag{11}$$

These loss functions are convex in z [2], [5], except for the sigmoid functions. The sigmoid-β function is technically not a surrogate loss function, we consider it for evaluation as for large values of β it approximates the $0-1$ loss function. Though hinge loss is a popular choice for loss function from a efficiency and performance perspective [19], a priori it is unclear which loss function will give the best results for a given dataset. Therefore we evaluate all these losses in our experiments.

IV. Similarity Learning

In this section, we introduce a specific family of similarity functions which our system employs. Let users and items be represented by vectors in $\mathbb{R}^d$. Let g be a function defined for a user x, item y, matrix A and item bias vector b, which returns the similarity between x and y and given by

$$g(x, y, A, b) = sim(x, y, A) + \frac{1}{1+e^{-b_y}}, \tag{12}$$

where b_y is the coordinate of b corresponding to item y. We pass the bias term through the sigmoid function to constrain its contribution. The item bias term captures the popularity of the item. As user bias terms cancel out in our risk functions $\mathscr{E}$, there is no bias term for users.

We extend the standard cosine similarity, which will take place of the sim function described above. Let $\mathbb{R}^d \to \mathbb{R}^p$ be a linear transformation defined by matrix $A \in \mathbb{R}^{p\times d}$. Consider the function

$$sim_{\text{cs}}(x, y, A) = \frac{x^T A^T A y}{\sqrt{x^T A^T A x}\sqrt{y^T A^T A y}}. \tag{13}$$

To treat the space of users and items separately, we introduce a variation of the above function, which transforms user and item vectors differently.

$$sim_{\text{cs}}(x, y, A, C) = \frac{x^T A^T C y}{\sqrt{x^T A^T A x}\sqrt{y^T C^T C y}} \tag{14}$$

The function g will play the role of function f_c introduced in the previous section and the matrices A and C and the bias vector for each channel will be learned from the data by optimizing empirical risk.

A. Optimization

Let J_ℓ denote the function obtained by replacing the indicator functions in the risk functions $\mathscr{E}$ by the loss function ℓ. There are regularization terms added to the function J_ℓ to prevent overfitting. The optimization problem is to find matrices, and vectors A_c, C_c, b_c and w_c for channels IP, OP and OB such that (dropping the channel subscript c)

$$\begin{aligned}\Theta = (A, C, b, w) &= \arg\min_{A,C,b,w} J_\ell(A, C, b, w) \\ &+\lambda_A\Omega[A] + \lambda_C\Omega[C] + \lambda_b\Omega[b] + \lambda_w\Omega[w],\end{aligned} \tag{15}$$

where $\lambda \in \mathbb{R}$ are the regularization parameters and $\Omega[.]$ is an appropriate matrix norm such as ℓ_F the Frobenius matrix norm. For an arbitrary matrix $A \in \mathbb{R}^{p\times d}$, it is defined as

$$\|A\|_F = \sum_{s=1}^{p}\sum_{t=1}^{d} A_{st}^2 \tag{16}$$

B. Algorithm

To learn matrices A, C and the bias vector b, the system performs a stochastic gradient descent (SGD) on the function J_ℓ. To implement SGD we make a minor change by replacing the set $\mathscr{I}_u^-$ with the following set

$$\mathscr{I}_{u,i}^* = \{j \in \mathscr{I}_u \mid B(u,i) > B(u,j)\}. \tag{17}$$

where $i \in \mathscr{I}_u^+$ as defined in the risk functions $\mathscr{E}$ and B is the consumption frequency matrix. This modification ensures that there is an ordering when consumed items are compared to each other. This alternative cost function minimizes the sum of ranks of the bought items in the recommended list, with items purchased more frequently having a higher rank than items purchased less frequently with highest rank being 0.

We exhibit some of the details required to perform the optimization in the special case when $\mathscr{E}_{MSL}$ Eq. (4), $\ell = \ell_\phi$ the sigmoid loss function and $A = A_{\text{IP}}$ is the matrix A for IP channel. Then the gradient of J with respect to A is given by

$$\nabla_A J_{\ell_\phi}(A) = \sum_{u,i,j} \nabla_A \ell_\phi(\Delta f(u,i,j)) \tag{18}$$

Further

$$\begin{aligned} \nabla_A \ell_\phi(\Delta f(u,i,j)) = -\ell_\phi(\Delta f(u,i,j))^2 \cdot \\ e^{\Delta f(u,i,j)} \cdot (\nabla_A f(u,i) - \nabla_A f(u,j)). \end{aligned} \tag{19}$$

The last term of Eq. (19) is determined by applying the quotient rule. Let v_1 and v_2 denote the numerator and denominator of sim_{cs}, then their respective gradients with respect to A are given by

$$\nabla_A v_1(x,y,A) = A(xy^T + yx^T) \tag{20}$$

$$\nabla_A v_2(x,y,A) = \frac{\sqrt{y^T A^T A y}}{\sqrt{x^T A^T A x}} Axx^T + \frac{\sqrt{x^T A^T A x}}{\sqrt{y^T A^T A y}} Ayy^T \tag{21}$$

The gradient computation with respect to C follows a similar procedure and with respect to b is simpler.

Algorithm 1 SGD iteration

```
1:  function SGD-ITERATION(𝒰, ℐ, Θ, w, cost)
2:      for u in 𝒰 do
3:          for i in ℐ_u^+ do
4:              for c in { IP, OP, OB } do
5:                  Compute f_c(u,i), ∇_Θ f_c(u,i)
6:              for j in ℐ*_{u,i} do
7:                  for c in { IP, OP, OB } do
8:                      Compute f_c(u,j), ∇_Θ f_c(u,j)
9:                  Update J_ℓ
10:                 Update Θ
11:                 Normalize w
12:     Update Θ, w                ▷ Regularization component
13:     Normalize w
14:     Update cost
15:     return Θ, w, cost
```

Algorithm 1 describes the structure of the optimization routine for our risk functions. The initial parameters to the algorithm are randomly chosen to be between 0 and 1. The weight vector w in Steps 11 and 13 of the algorithm is normalized such that the absolute values of the channel weights sum to 1.

C. Time Complexity

Each iteration of stochastic gradient descent described in Algorithm 1 requires

$$O(\sum_u \#\mathscr{I}_u^+ \cdot \#\mathscr{I}_u^-)$$

number of update operations.

There are a couple of techniques which will practically speed up the optimization. We merely mention them as we have not extensively investigated their impact on performance of our system. First, items which have never been purchased in the training phase for each channel can be filtered out. One choice for bias term for these items could be the average item bias. Implementing this technique requires some amount of book-keeping as training is restricted to certain set of items while testing is not. Second, the two `for` loops which iterate over each channel in Algorithm 1 can be parallelized as each channel computation is independent of the others. Third, given that a user u consumes only a few items the `for` loop in line 6 of the above algorithm could iterate over a random sample of $\mathscr{I}_u^-$ rather the iterating over the entire set.

V. Experiment

We describe our experimental setup and evaluate our system's performance in this section. The regularization parameters λ_A, λ_C, λ_b and λ_w from Eq. (15) are equal to λ, which takes on values $0.0011, 0.0031$ and 0.0101. In the SGD procedure, the learning rate for matrices A, C and b is α and the rate for the weight vector w is $0.01 \cdot \alpha$. The parameter α in the iteration number t of SGD is equal to $e^{-2 \cdot t}$. The SGD procedure terminates when the absolute value of the difference in cost between iterations drops below 0.001.

A. Dataset

Our dataset is from a retail chain with physical stores and an E-commerce website. The data encompasses customer transactions over 225 days period. The dataset consists of around 256 million transactions, distributed over about 2 million customers. All the items in the dataset can be aggregated into 700 item categories. All the items in a particular category share the same features.

Our choice of category is driven by the following factors: the category has a large number of items, the items are bought frequently in the dataset, the purchase cycle of items is short and items have good descriptions so that features could be extracted based on the item descriptions.

We narrowed our dataset to 10 such categories consisting of about 30 million transactions. We showcase results for one such category which has about 1000 items and a filtered set of 2000 customers with the average size of the set $\mathscr{I}_u^+$ being 6.29 and standard deviation being 4.74 for the IP channel.

B. Item and User profiles

The item descriptions consist of the following features: Price – numeric, Brand – categorical, Color and other category specific features, which were both numeric and categorical. All these features are converted into binary values resulting

in items being represented by boolean vectors in $\mathbb{R}^d$, where $d = 418$. This is done by introducing a feature for all possible values of categorical variables and binning the numeric variables. This boolean vector is referred to as item profile.

For each user in the dataset, user profile is generated by aggregating the boolean vectors of items consumed (purchased or bought) by that user, and the time of consumption is dictates whether the vector is being generated for the purpose of training or testing. This is discussed in the next subsection.

C. Train and Test Set Generation

The dataset of around 225 days is divided into two parts : two-thirds for training and one-third for testing. Further, the training set spanning over first 150 days is subdivided into two subparts — the first 125 days are used to create the user profile vectors and eventually similarities. The remaining 25 days are used to create a binary vector to indicate purchase of the item by the user in that period, and this acts as the target variable. Similarly for the test set, the first 200 days were used to create the user profile, while the remaining 25 days were used to create the target variable. Our train and test sets have 880 users each. Each SGD computation selects a random subset of the customers of size 200 as the training set.

D. Evaluation Metrics

Given that we view the problem of recommendation as a learning to ranking problem, ranking metrics are a natural choice for the purpose of evaluation of the system [13]. In our context, relevance is a binary function indicating whether the item was purchased or not and $@k$ denotes a recommendation list consisting of items with the first k ranks. We set $k = 10$ borrowing from search engine literature. In the results sections, the following metrics are presented: AUC is the probability that a randomly chosen item purchased in-store is ranked higher than one which is not, Recall$@k$ is the fraction of relevant items that are recommended, Precision$@k$ is the fraction of recommended items that are relevant, nDCG$@k$ is a score based on the graded relevance of recommended items and Average Precision (AVEP) is the Precision$@n$ of a ranking of items, averaged over all positions n of purchased items. The tables showcase these metrics averaged over the users.

Given that our approach is to minimize rank loss, AUC results are of primary importance in our experiments. From the perspective of recommendation systems, recall is also a metric of importance. We include nDCG and AVEP results as they are standard metrics in the ranking literature. We highlight the best AUC result in each of our experiments in the next subsection.

E. Baseline Results

Table I displays our baseline results where channel-specific scores computed using standard cosine similarity are summed and the top-k scores determine the recommendations for IP. This is done for individual and combinations of channels. The table gives evidence to the hypothesis that increasing the number of channel similarity scores does not imply an increase in the evaluation metrics as is evident when moving from 2 channels (IP, OP) to 3 channels (IP, OP, OB). This is due to the OB channel profiles being sparse and not being a good source of prediction.

We note that to our knowledge there are no publicly available multi-channel datasets with which we can evaluate our system; this limits the results that we can publicly share.

F. Our Results

We now proceed to showcase the results of our approach. The label sets $\mathscr{I}_u^+$ and $\mathscr{I}_u^-$ referenced in Eqs. 4, 5 and 6 (MLSI, MLST and MLSH) refer to consumption activity in the in-store purchase (IP) channel. These labels are from two durations: the first 125 days and days 126-150 in our dataset. We refer to the first scenario as IP-1-125 and the second scenario as IP-126-150.

Tables II and III provide results for MLSH and MLST with labels from IP-1-125 using the similarity function in Eq. 13 with $\lambda = 0.0011$. Given that a single matrix A is being learned for each channel, the costs in the associated optimization are extremely high and the SGD procedure takes a long time to terminate. These tables suggest that the results are around the three channel baseline AUC value implying tranforming users and items separately by learning matrices A and C for the similarity function in Eq. 14 is a better approach as later tables will demonstrate.

Tables IV and V arrange data for MLSH with labels from IP-126-150 and IP-1-125 respectively. For these tables we compute evaluation metrics for selective combinations of the regularization parameter $\lambda = 0.0011$, 0.0031 and 0.0101 and values of $p = 5, 15$ and 25. The AUC results of the latter table are around the AUC value for the three channel baseline (0.8458) from Table I while the larger duration label set IP-1-125 has significantly better AUC values.

Tables VI, VII, VIII contrast the models MLSH, MLST and MLSI using three channels scores (IP, OP, and OB) and using two channel scores (IP and OP). In each of these tables rows with same values of p and loss l use the same set of customers and are hence comparable. Consistent with the baseline AUC results, two channel AUC results fare better than three channel ones. It is difficult to differentiate the performance of these models and therefore a need to perform more theoretical analysis and experiments.

The best AUC result of 0.9301 in our experiments is obtained by the MLSI function with labels from the first 125 day period (Table IX) using scores from the IP and OP channels. This is a significant improvement over the two channel baseline AUC value of 0.8508 from Table I. In this experiment labels do not come from the target channel as in other experiment but from the respective channels. One might consider a second phase of learning to make the results sensitive to the target channel.

In Table X we compare results for one versus two channels with labels from the IP-1-125 duration. The first subtable refers using IP channel scores, where the three risk functions are equivalent. The subtables which follow are for MLSI, MLST and MLSH. This table is visualized in Figure 2. The number of customers chosen for these results were 100. This table shows that on average using two channels of IP and OP is better than merely using one channel of IP for MLST and MLSH but not for MLSI.

Channels	AUC	Recall	Precision	nDCG	AVEP
IP	0.8355	0.3932	0.1306	0.3331	0.2558
OP	0.8106	0.3732	0.1220	0.2882	0.2161
OB	0.3075	0.0968	0.0299	0.0784	0.0520
IP, OP	**0.8508**	0.4022	0.1332	0.3385	0.2632
IP, OP, OB	0.8458	0.3819	0.1255	0.3181	0.2440

TABLE I. RESULTS FOR STANDARD COSINE CHANNELS FOR COMBINATIONS OF CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	**0.8472**	0.1709	0.056	0.1151	0.0679	0.3346	0.3332	0.3322
5	$+$	0.0011	0.8349	0.1686	0.06	0.1019	0.0543	0.3345	0.3797	0.2858
5	log	0.0011	0.8453	0.1979	0.0665	0.1423	0.0891	0.3321	0.3334	0.3345
5	ϕ_1	0.0011	0.8119	0.0962	0.0306	0.0692	0.0422	0.2703	0.2718	0.4579

TABLE II. RESULTS FOR MLSH WITH LABELS FROM IP-1-125 AND A=C WITH SCORES FROM THE IP, OP AND OB CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	**0.8766**	0.1983	0.0682	0.1444	0.0892	0.3393	0.3093	0.3514
5	$+$	0.0011	0.8271	0.1024	0.032	0.0588	0.0266	0.3807	0.3744	0.2449
5	log	0.0011	0.8437	0.194	0.0594	0.1465	0.0911	0.2874	0.381	0.3316
5	exp	0.0011	0.8529	0.2206	0.0735	0.1571	0.0988	0.3376	0.6036	0.0589
5	ϕ_1	0.0011	0.8671	0.2277	0.0763	0.1649	0.1061	0.3091	0.3699	0.3209

TABLE III. RESULTS FOR MLST WITH LABELS FROM IP-1-125 AND A=C WITH SCORES FROM THE IP, OP AND OB CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	0.8553	0.1541	0.0611	0.1084	0.0694	0.3343	0.3392	0.3266
5	ϕ	0.0031	0.825	0.1588	0.0531	0.1036	0.0609	0.3591	0.329	0.3119
5	ϕ	0.0101	0.8435	0.2158	0.0638	0.1527	0.0907	0.3264	0.3263	0.3474
5	$+$	0.0011	0.8482	0.194	0.0618	0.1248	0.0712	0.2949	0.3102	0.3949
5	$+$	0.0031	0.8557	0.1888	0.0697	0.1171	0.0642	0.3509	0.3788	0.2702
5	$+$	0.0101	0.8382	0.1139	0.0441	0.0741	0.0409	0.3423	0.3096	0.3482
5	log	0.0011	0.8622	0.2002	0.0638	0.1281	0.0737	0.4147	0.2847	0.3006
5	log	0.0031	0.8488	0.1257	0.0478	0.0746	0.0365	0.31	0.2834	0.4066
5	log	0.0101	0.8366	0.0896	0.0383	0.062	0.0331	0.3433	0.3527	0.304
5	exp	0.0011	0.8606	0.1158	0.0468	0.0938	0.0591	0.337	0.3133	0.3497
5	exp	0.0031	0.8324	0.1055	0.0392	0.0691	0.0337	0.3604	0.3068	0.3328
5	exp	0.0101	0.8383	0.1588	0.0545	0.1024	0.0536	0.3124	0.2841	0.4035
5	ϕ_1	0.0011	0.8418	0.1907	0.0616	0.1248	0.0711	0.3161	0.3434	0.3404
5	ϕ_1	0.0031	0.8522	0.1925	0.0544	0.1389	0.0842	0.3244	0.3442	0.3314
5	ϕ_1	0.0101	0.8327	0.1287	0.0519	0.0863	0.0448	0.2619	0.367	0.3711
5	ϕ_{10}	0.0011	0.8513	0.1875	0.0617	0.1024	0.0479	0.1478	0.3833	0.4689
5	ϕ_{10}	0.0031	0.8413	0.1889	0.0615	0.1466	0.0912	0.2779	0.3588	0.3633
5	ϕ_{10}	0.0101	**0.8809**	0.2113	0.0749	0.1319	0.0692	0.4069	0.3385	0.2546
15	ϕ	0.0011	0.8292	0.191	0.0644	0.1249	0.0767	0.3437	0.3422	0.3142
15	$+$	0.0011	0.8329	0.1776	0.0631	0.1092	0.0597	0.3851	0.3315	0.2835
15	log	0.0011	0.8265	0.1835	0.0616	0.1098	0.0605	0.3695	0.276	0.3544
15	exp	0.0011	0.8597	0.1658	0.0616	0.1127	0.069	0.3178	0.332	0.3501
15	ϕ_1	0.0011	0.8708	0.2247	0.0726	0.1427	0.0798	0.3326	0.3407	0.3267
15	ϕ_{10}	0.0011	0.84	0.1662	0.057	0.0949	0.0468	0.3336	0.3465	0.32
25	ϕ	0.0011	0.823	0.1954	0.0659	0.1363	0.0794	0.3365	0.3248	0.3388
25	$+$	0.0011	0.8584	0.1626	0.0601	0.1114	0.0684	0.3697	0.2982	0.3321
25	log	0.0011	0.8322	0.1733	0.0613	0.1103	0.0575	0.3224	0.3691	0.3085
25	exp	0.0011	0.8587	0.2119	0.0689	0.1472	0.0887	0.3248	0.3536	0.3215

TABLE IV. RESULTS FOR MLSH WITH LABELS FROM IP-126-150 WITH SCORES FROM THE IP, OP AND OB CHANNELS

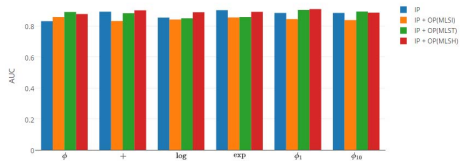


Fig. 2. AUC values across various losses for one versus two channels

VI. PRODUCT QUOTIENT

Retailers and publishers would like to score items in their inventory to understand the significance of each item to their business from a consumption perspective. These items could be products or articles and consumption could mean purchase or browse behavior depending on the context. For the purpose of this discussion, we will specialize to the retail scenario. We propose two natural notions of a product's significance:

- *Self-promotion value* captures the effect of the sale of the product on itself. This can be viewed as repeat purchase propensity of the item.
- *Cross-promotion value* influences the sale of other products. This quantifies the cross sell opportunity of the product.

These values can be viewed as the product's equity or the collective valuation of the product by consumers. Using these product values to quantify the store's performance is

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	0.8662	0.1915	0.0677	0.1393	0.0889	0.3343	0.3392	0.3266
5	ϕ	0.0031	0.856	0.1796	0.0634	0.1192	0.0697	0.3591	0.329	0.3119
5	ϕ	0.0101	0.8588	0.1478	0.0524	0.1	0.0562	0.3264	0.3263	0.3474
5	$+$	0.0011	0.8583	0.1494	0.0502	0.0868	0.047	0.2949	0.3102	0.3949
5	$+$	0.0031	0.8662	0.1948	0.0643	0.1225	0.071	0.3509	0.3788	0.2702
5	$+$	0.0101	0.8683	0.1751	0.0677	0.122	0.0724	0.3423	0.3096	0.3482
5	log	0.0011	0.8644	0.1565	0.0573	0.0942	0.0468	0.4147	0.2847	0.3006
5	log	0.0031	0.8659	0.1522	0.0602	0.0953	0.0575	0.31	0.2834	0.4066
5	log	0.0101	0.8585	0.1331	0.0503	0.0955	0.0586	0.3433	0.3527	0.304
5	exp	0.0011	0.8501	0.0898	0.0325	0.0485	0.0207	0.337	0.3133	0.3497
5	exp	0.0031	0.8761	0.1333	0.0495	0.0802	0.0381	0.3604	0.3068	0.3328
5	exp	0.0101	0.898	0.2674	0.083	0.1837	0.1158	0.3124	0.2841	0.4035
5	ϕ_1	0.0011	0.8817	0.2492	0.0781	0.1637	0.1002	0.3161	0.3434	0.3404
5	ϕ_1	0.0031	0.8726	0.1664	0.0603	0.1107	0.0608	0.3244	0.3442	0.3314
5	ϕ_1	0.0101	0.864	0.1548	0.0534	0.0934	0.0489	0.2619	0.367	0.3711
5	ϕ_{10}	0.0011	**0.9039**	0.2652	0.092	0.1793	0.1084	0.1478	0.3833	0.4689
5	ϕ_{10}	0.0031	0.8672	0.1666	0.0598	0.1146	0.0718	0.2779	0.3588	0.3633
5	ϕ_{10}	0.0101	0.8872	0.18	0.0709	0.1225	0.073	0.4069	0.3385	0.2546
15	ϕ	0.0011	0.8912	0.2362	0.0759	0.1598	0.0989	0.3437	0.3422	0.3142
15	$+$	0.0011	0.8555	0.0848	0.0376	0.0512	0.024	0.3851	0.3315	0.2835
15	log	0.0011	0.8955	0.2335	0.0764	0.1503	0.0867	0.3695	0.276	0.3544
15	exp	0.0011	0.8842	0.2227	0.0664	0.1492	0.0844	0.3178	0.332	0.3501
15	ϕ_1	0.0011	0.8853	0.2117	0.0747	0.1438	0.0931	0.3326	0.3407	0.3267
15	ϕ_{10}	0.0011	0.8934	0.2743	0.0901	0.1789	0.1184	0.3336	0.3465	0.32
25	ϕ	0.0011	0.8747	0.2303	0.0768	0.1474	0.0855	0.3365	0.3248	0.3388
25	$+$	0.0011	0.8993	0.2528	0.0808	0.1623	0.1024	0.3697	0.2982	0.3321
25	log	0.0011	0.8911	0.2853	0.0948	0.185	0.1139	0.3224	0.3691	0.3085
25	exp	0.0011	0.89	0.1708	0.0642	0.1156	0.063	0.3248	0.3536	0.3215

TABLE V. RESULTS FOR MLSH WITH LABELS FROM IP-1-125 WITH SCORES FROM THE IP, OP AND OB CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	0.8597	0.2143	0.0642	0.1456	0.0856	0.3149	0.3567	0.3283
5	$+$	0.0011	0.8574	0.1317	0.0535	0.0911	0.0555	0.3133	0.3594	0.3273
5	log	0.0011	0.8774	0.2145	0.0677	0.1447	0.09	0.3778	0.2962	0.326
5	exp	0.0011	0.8867	0.1238	0.0474	0.0892	0.051	0.4317	0.2605	0.3078
5	ϕ_1	0.0011	0.8713	0.1599	0.0578	0.1058	0.0572	0.3486	0.3302	0.3212
5	ϕ_{10}	0.0011	**0.8919**	0.2455	0.0851	0.1561	0.0918	0.4961	0.1348	0.3691
5	ϕ	0.0011	0.8947	0.1931	0.0664	0.1212	0.0708	0.4262	0.5738	0.0
5	$+$	0.0011	**0.9175**	0.2762	0.0906	0.191	0.1181	0.5092	0.4908	0.0
5	log	0.0011	0.8898	0.1565	0.0574	0.1137	0.0681	0.5426	0.4574	0.0
5	exp	0.0011	0.9015	0.2338	0.0795	0.1621	0.0966	0.6821	0.3179	0.0
5	ϕ_1	0.0011	0.8878	0.1305	0.0501	0.0949	0.0565	0.3459	0.6541	0.0
5	ϕ_{10}	0.0011	0.9167	0.2418	0.0838	0.1617	0.0974	0.6619	0.3381	0.0

TABLE VI. RESULTS FOR MLSH WITH LABELS FROM IP-1-125 WITH SCORES FROM THE IP, OP AND OB CHANNELS VERSUS SCORES FROM THE IP AND OP CHANNELS

a related business problem. In this invention, we propose a scoring function that will identify a product's self-promotion and cross-promotion values more accurately over traditional methods.

We define two *product quotient* functions: $\text{PQ}^{\text{cross}}(i, j)$ and $\text{PQ}(i)$. The first function captures the effect of the purchase of one unit of item i on the purchase of item j. The second function encodes the effect of the purchase of one unit of item i on the purchase of item i; this can be viewed as a repeat purchase propensity of item i and is a special case of the first function. These two functions are realizations of the promotional values of items introduced earlier.

The definition of the similarity function introduced in section IV involves a user x and item y, which share the same feature space (see section V for a list of features). The item vector is a binary vector, whereas the user vector is an accumulation over the items consumed by the user. In what follows, we replace the user vector x by an item vector i to introduce functions which compare two products via the similarity function. This should be viewed as a user who consumes one unit of item i.

We define two *product quotient* functions: $\text{PQ}^{\text{cross}}(i, j)$ and $\text{PQ}(i)$. The first function captures the effect of the purchase of one unit of item i on the purchase of item j

$$\text{PQ}^{\text{cross}}(i, j) = \text{sim}_{\text{cs}}(i, j, A, C) \tag{22}$$

where A and C are learned from the data. As the first parameter in the similarity function refers to a user profile, $\text{PQ}^{\text{cross}}(i, j)$ can be viewed as how the purchase of one unit of item i influences the purchase of one unit of item j. As the matrices A and C are learned from the data this relationship is likely to be asymmetric in general.

The second function encodes the effect of the purchase of one unit of item i on the purchase of item i; this can be viewed as a repeat purchase propensity of item i. The function PQ is a special case of the PQ^{cross} function, specifically

$$\text{PQ}(i) = \text{PQ}^{\text{cross}}(i, i) \tag{23}$$

We define the value of the store S based on consumption behavior to be the sum of item quotients, refer to it as the *retailer quotient* and denote it by RQ, where the set S is the colllection of item ($\mathscr{I}$) and user profiles $\mathscr{U}$.

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	0.8377	0.0549	0.0175	0.038	0.0211	0.3634	-0.1778	0.4588
5	$+$	0.0011	0.8246	0.1359	0.0433	0.0859	0.0451	-0.1298	0.498	0.3722
5	log	0.0011	0.8142	0.1916	0.0625	0.128	0.0732	-0.4049	0.5042	0.0909
5	exp	0.0011	0.7432	0.0967	0.0365	0.0569	0.0272	-0.0042	0.2901	0.7057
5	ϕ_1	0.0011	0.8622	0.2175	0.0758	0.1364	0.0793	0.3236	0.3878	0.2887
5	ϕ_{10}	0.0011	**0.8793**	0.2054	0.0628	0.1427	0.0817	0.4894	0.3209	0.1898
5	ϕ	0.0011	0.8978	0.2063	0.0667	0.1361	0.0773	0.4662	0.5338	0.0
5	$+$	0.0011	0.8901	0.189	0.0642	0.1322	0.0777	0.5434	0.4566	0.0
5	log	0.0011	0.9187	0.2061	0.0759	0.143	0.0861	0.6021	0.3979	0.0
5	exp	0.0011	0.9008	0.2118	0.0703	0.1262	0.0678	0.6809	0.3191	0.0
5	ϕ_1	0.0011	0.9081	0.2233	0.0747	0.1663	0.1075	0.4997	0.5003	0.0
5	ϕ_{10}	0.0011	**0.9216**	0.3117	0.1067	0.2045	0.1295	0.6154	0.3846	0.0

TABLE VII. RESULTS FOR MLST WITH LABELS FROM IP-1-125 WITH SCORES FROM THE IP, OP AND OB CHANNELS VERSUS SCORES FROM THE IP AND OP CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP	w_{IP}	w_{OP}	w_{OB}
5	ϕ	0.0011	**0.8918**	0.1706	0.058	0.1052	0.0577	0.334	0.333	0.333
5	$+$	0.0011	0.8761	0.1158	0.0503	0.0793	0.043	0.334	0.333	0.333
5	log	0.0011	0.8682	0.1969	0.0728	0.1262	0.0761	0.334	0.333	0.333
5	exp	0.0011	0.8818	0.084	0.0318	0.0503	0.0245	0.334	0.333	0.333
5	ϕ_1	0.0011	0.8914	0.1893	0.056	0.133	0.0782	0.334	0.333	0.333
5	ϕ_{10}	0.0011	0.8912	0.239	0.084	0.1583	0.0941	0.334	0.333	0.333
5	ϕ	0.0011	0.9088	0.2233	0.0786	0.158	0.1041	0.5	0.5	0.0
5	$+$	0.0011	0.8952	0.193	0.0683	0.1376	0.0812	0.5	0.5	0.0
5	log	0.0011	0.9199	0.246	0.0849	0.1697	0.1031	0.5	0.5	0.0
5	exp	0.0011	**0.9216**	0.2913	0.0993	0.2135	0.1391	0.5	0.5	0.0
5	ϕ_1	0.0011	0.9177	0.2199	0.0747	0.1604	0.1043	0.5	0.5	0.0
5	ϕ_{10}	0.0011	0.9205	0.2783	0.0999	0.1951	0.1283	0.5	0.5	0.0

TABLE VIII. RESULTS FOR MLSI WITH LABELS FROM IP-1-125 WITH SCORES FROM THE IP, OP AND OB CHANNELS VERSUS SCORES FROM THE IP AND OP CHANNELS

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP
5	$+$	0.0011	0.9173	0.2591	0.092	0.1666	0.097
5	exp	0.0011	0.9106	0.2068	0.0733	0.131	0.0711
5	ϕ_{10}	0.0011	**0.9301**	0.3113	0.1106	0.2206	0.1458

TABLE IX. RESULTS FOR MLSI WITH LABELS FROM THE RESPECTIVE CHANNELS AND THE 1-125 DAY PERIOD USING SCORES FROM THE IP AND OP CHANNELS

$$\text{RQ}(S) = \frac{\sum_{i \in \mathscr{I}} \text{PQ}^{\text{cross}}(i,i)}{\#\mathscr{I}} \tag{24}$$

Note that since this score is normalized by the number of products, it is insensitive to the size of the inventory; this enables comparison between retailers. Lastly, we remark that comparing our product scoring system against competing systems could be pursued in the future.

A. Results

In this section we will provide evidence to support our hypothesis that our scoring system which captures consumption behavior is better than a system which uses cosine similarity. For a fixed item i, let $\text{rankPQ}^{\text{cross}}(i,j)$ denote the rank of $\text{PQ}^{\text{cross}}(i,j)$ j ranges over the list of all items. The rank $\text{ranksim}_{\text{cs}}(i,j)$ has an analogous definition for cosine similarity, where $A = C$ is the identity matrix.

We first share results on the PQ^{cross} function. Let B be the set of 50 most frequently bought items in our data set. And for every pair of items i and j in B we computed the values of $\text{PQ}^{\text{cross}}(i,j)$. The number of times $\text{rankPQ}^{\text{cross}}(i,j)$ is better than $\text{ranksim}_{\text{cs}}(i,j)$ is 2312 out of 2500. This indicates that frequently bought items promote each other according to our product scoring system. Next we also examined items which were rarely bought (long tail of the purchased items distribution) and these items were 50 in number; let's label this set R. In this case, the number of times $\text{rankPQ}^{\text{cross}}(i,j)$ is better than $\text{ranksim}_{\text{cs}}(i,j)$ is 903 out of 2500. The ranks for this set of items did not exhibit behavior demonstrated by the items in set B. Figures 5, 6 plot how the most and least frequently purchased items (mo and le) influence the top-50 (set B) and bottom-50 (set R) respectively. The vertex label in these graphs specify $\text{PQ}^{\text{cross}}(mo, j_1)$ and $\text{PQ}^{\text{cross}}(le, j_2)$, where $j_1 \in B$ and $j_2 \in R$.

Next we look at the distribution of the rankPQ(i,i) function for top-50 and bottom-50 frequently purchased items (Figures 3, 4). The histograms are right-skewed and left-skewed which correlates with purchase frequency. Recall that traditional cosine similarity will give the same score of 1 to all items resulting inability to differentiate between the ranks. This documents the superiority of our product quotient system. Finally the retailer quotient of this company was calculated to be 0.733.

Our scoring system captures the significance of the item in terms of how a purchase of this item influences the value of the store via promotional values. Furthermore, we define the retailer's value based on item values. To our knowledge, no existing system scores the items in this way. This method of learning similarities from the transaction history provides a better measure of item scoring than standard cosine similarity measure on transaction history. This is because in the latter,

p	ℓ	λ	AUC	Recall	Precision	nDCG	AVEP
5	ϕ	0.0011	0.8312	0.22	0.0756	0.1504	0.0933
5	$+$	0.0011	0.8917	0.1724	0.0632	0.1114	0.0599
5	log	0.0011	0.8539	0.0492	0.0203	0.0284	0.0109
5	exp	0.0011	**0.9021**	0.2344	0.079	0.1639	0.1022
5	ϕ_1	0.0011	0.884	0.2278	0.0759	0.1568	0.0995
5	ϕ_{10}	0.0011	0.8841	0.119	0.0494	0.0921	0.0569
5	ϕ	0.0011	**0.8575**	0.1902	0.0645	0.1253	0.0698
5	$+$	0.0011	0.8321	0.1331	0.0418	0.0665	0.0268
5	log	0.0011	0.8418	0.1116	0.0422	0.0745	0.0397
5	ϕ_1	0.0011	0.8442	0.2129	0.0713	0.1462	0.0942
5	ϕ_{10}	0.0011	0.8375	0.2137	0.0716	0.1484	0.0936
5	ϕ	0.0011	0.8903	0.2421	0.0799	0.172	0.1139
5	$+$	0.0011	0.8825	0.1494	0.0555	0.0926	0.0484
5	log	0.0011	0.8493	0.1236	0.0457	0.0765	0.037
5	exp	0.0011	0.857	0.1788	0.0626	0.1158	0.0748
5	ϕ_1	0.0011	**0.9033**	0.2454	0.0872	0.1784	0.1223
5	ϕ_{10}	0.0011	0.8928	0.2636	0.0915	0.1796	0.1135
5	ϕ	0.0011	0.8771	0.199	0.07	0.1375	0.0932
5	$+$	0.0011	0.9012	0.1696	0.0614	0.0987	0.0488
5	log	0.0011	0.8883	0.2012	0.0588	0.1508	0.0953
5	exp	0.0011	0.8908	0.2031	0.0708	0.1291	0.0704
5	ϕ_1	0.0011	**0.9088**	0.2272	0.0713	0.1578	0.0951
5	ϕ_{10}	0.0011	0.8858	0.2134	0.0765	0.14	0.0916

TABLE X. RESULTS FOR ONE CHANNEL VERSUS RESULTS FOR MLSI, MLST, MLSH FOR THE IP AND OP CHANNELS WITH LABELS FROM IP-1-125

Fig. 3. Distribution of rankPQ of top-50 frequently purchased items

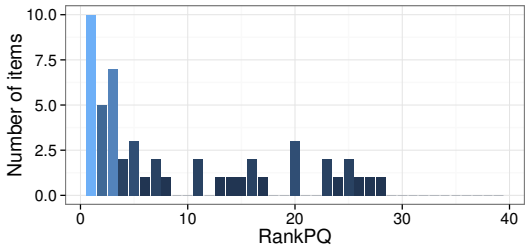
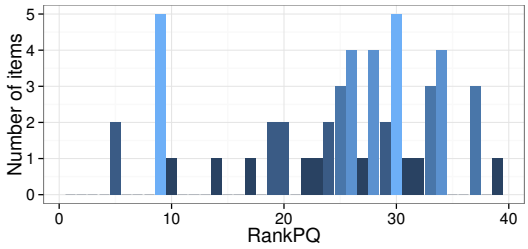


Fig. 4. Distribution of rankPQ of bottom-50 frequently purchased items



an item promotes itself the most, independent of the nature of the data (in a cosine similarity based scoring model, item self-promotion scores would be always be 1), whereas this might not always be the case. In the context of cross-promotional values, our method is flexible to accommodate asymmetry in promotionality, that is, the purchase of item i driving the sales of item j does not imply the reverse direction promotion. This is in contrast to using standard cosine similarity for scoring due to its symmetric nature. This illustrates that our method provides a more accurate picture of the value of items.

B. Applications

Applications of the product quotient system include new ways to evaluate retailer performance, design products in the retail context. We elaborate on two of these applications.

Fig. 5. A graph describing how the most frequently bought product influences the top-50 frequently bought products

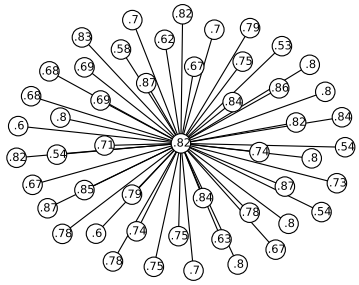
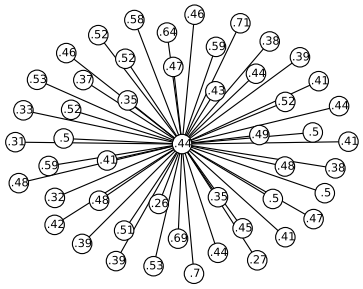


Fig. 6. A graph describing how the least frequently bought product influences the bottom-50 frequently bought products



The learned similarity measure can be used in designing a product having maximum promotional values. This is achieved by performing automated variable selection techniques via a greedy solution. The algorithm would ensure that only one feature is selected from binary features corresponding to a class of features, for instance, exactly one brand is selected. A special instance of this application is the following: suppose the product features have been finalized but one feature, price for instance, is yet to be fixed. The price range which maximizes the item's value can be computed by keeping all but

price features fixed and iteratively selecting one price feature at a time. This is related to the technique of conjoint analysis which attempts to answers what features a new product should have and how it should be priced.

Another application of our system is to identify a set of products which collectively drives a larger set of products; this is valuable information to the retailer. Consider the set of products as vertices of a graph with a directed edge from product i and j if the former drives the latter with weight of the edge being equal to $\text{PQ}^{\text{cross}}(i, j)$. Using the concept of network flows, the set of promoting items can be determined. (A dummy source and sink with infinite capacity edges might need to be added.)

VII. Conclusion and Future work

In this paper we have introduced a recommendation system using the pair-wise learning to rank approach, where *rank loss* is minimized using surrogate loss functions. Our approach of learning similarity functions gives significant performance improvement over using standard cosine similarity particularly when distinct user and item transformation matrices are learned using a multi-channel marketing dataset.

As future work, it will be interesting to investigate the performance of other similarity and distance measures [7] in place of cosine similarity. A feature of our system is that it is agnostic to the choice of target channel; this is another avenue of exploration. Finally, the phenomenon of similarity scores from fewer channels give better performance spells out the need to couple automatic feature selection with our learning algorithm in applications where there are many channels to consider.

Our algorithm is designed to maximize AUC and it appears that large AUC values are accompanied by large values for the other metrics. Comments on this correlation will have to be based on further analysis. AUC results tend to improve as p the number of rows in the learned matrices increases. Given that it is computationally expensive to evaluate the performance of the system for various values of p, it would be worthwhile to devise an optimization algorithm which determines a value of p which maximizes AUC and computes the associated matrices.

Acknowledgements

We would like to thank the members of Adobe Research, Bangalore, in particular Shriram Revankar, Ritwik Sinha and Shiv Saini for their feedback.

References

[1] S. Balakrishnan and S. Chopra. Collaborative ranking. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 143–152, New York, NY, USA, 2012. ACM.

[2] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. (Was Department of Statistics, U.C. Berkeley Technical Report number 638, 2003).

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual conference on Computational Learning Theory*, pages 92–100. ACM, 1998.

[4] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[5] W. Chen, T.-Y. Liu, Y. Lan, Z. Ma, and H. Li. Ranking measures and loss functions in learning to rank. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS*, pages 315–323. Curran Associates, Inc., 2009.

[6] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA, 2001. ACM.

[9] W. Kotlowski, K. Dembczynski, and E. Huellermeier. Bipartite ranking through minimization of univariate loss. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1113–1120, New York, NY, USA, June 2011. ACM.

[10] J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer. Local collaborative ranking. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 85–96, New York, NY, USA, 2014. ACM.

[11] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM, 2008.

[12] B. McFee, L. Barrington, and G. Lanckriet. Learning content similarity for music recommendation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2207–2218, 2012.

[13] B. McFee and G. Lanckriet. Metric learning to rank. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 775–782, Haifa, Israel, June 2010. Omnipress.

[14] B. McFee and G. Lanckriet. Learning multi-modal similarity. *The Journal of Machine Learning Research*, 12:491–523, 2011.

[15] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part II*, pages 709–720, 2010.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[17] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *International Joint Conference on Artificial Intelligence*, volume 22, page 2318, 2011.

[18] C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127, 2014.

[19] L. Rosasco, E. De, V. A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16:1063–1076, 2004.