Non-Linear Label Ranking for Large-Scale Prediction of Long-Term User Interests

Nemanja Djuric[†], Mihajlo Grbovic[†], Vladan Radosavljevic[†], Narayan Bhamidipati[†], Slobodan Vucetic[‡]

[†]Yahoo! Labs, Sunnyvale, CA, USA, {nemanja, mihajlo, vladan, narayanb}@yahoo-inc.com

[‡]Temple University, Philadelphia, PA, USA, vucetic@temple.edu

Abstract

We consider the problem of personalization of online services from the viewpoint of ad targeting, where we seek to find the best ad categories to be shown to each user, resulting in improved user experience and increased advertisers' revenue. We propose to address this problem as a task of ranking the ad categories depending on a user's preference, and introduce a novel label ranking approach capable of efficiently learning non-linear, highly accurate models in large-scale settings. Experiments on a real-world advertising data set with more than 3.2 million users show that the proposed algorithm outperforms the existing solutions in terms of both rank loss and top-K retrieval performance, strongly suggesting the benefit of using the proposed model on large-scale ranking problems.

Introduction

Personalization of online content has become an important topic in the recent years. It has been defined as "the ability to proactively tailor products and product purchasing experiences to tastes of individual consumers based upon their personal and preference information" (Chellappa and Sin 2005), which may lead to improved user experience and directly translate into financial gains for online businesses (Riecken 2000). In addition, personalization fosters stronger bond between users and companies, and can help in increasing user loyalty and retention (Alba et al. 1997). For these reasons it has been recognized as an important strategic goal of major internet companies (Manber, Patel, and Robison 2000; Das et al. 2007), and is a focus of significant research efforts. Personalized content has already become an integral part of many popular online services, a trend likely to continue in the future (Tuzhilin 2009).

We consider content personalization from the viewpoint of targeted advertising (Essex 2009), an increasingly important aspect of online businesses. Here, for each individual user the task is to find the best matching ads to be displayed, which improves user's online experience (as only relevant and interesting ads are shown to the user) and can lead to increased revenue for the advertisers (as users are more likely to click on the ad and make a purchase). Due to its large impact and many open research questions, targeted advertising has garnered significant interest from the machine learning community, as witnessed by a large number of recent workshops and publications (Broder 2008; Pandey et al. 2011; Majumder and Shrivastava 2013).

One of the most popular approaches in present-day targeting, particularly in brand awareness campaigns, is to assign categories to the display ads, such as "sports" or "finance", and then separately learn to predict user interest in each of these categories using historical records (Ahmed et al. 2011; Pandey et al. 2011; Tyler et al. 2011). Typically, a taxonomy is used to decide on the categories, and depending on how detailed it is hundreds of separate category qualification tasks may need to be solved. Thus, for each ad category, a separate predictive model is trained, able to estimate the probability of an ad click for the entire user population. Then, for each category, N users with the highest click probability are selected for ad exposure. Known issues with the approach include overexposure, where a single user may be among the top N users for many categories, and starvation, where some users do not qualify for any of the categories.

An alternative avenue, known in the industry as a user-interest model, is to sort for each user outputs of the predictive models, and qualify users based on their top K categories. The approach guarantees that a user is qualified into several categories, eliminating overexposure and starvation issues. However, this method may still be suboptimal, as the predictive models are trained in isolation and do not consider relationships between different categories. In this paper we explore methods capable of capturing more complex class dependencies, and consider the user-interest model from a label ranking standpoint (Vembu and Gärtner 2011). However, the sheer scale of ad targeting problems, with data sets comprising millions of users and features and hundreds of categories, renders many existing label ranking approaches intractable, presenting new challenges to the researchers.

To address this issue, we propose a novel label ranking algorithm suitable for large-scale settings. The method lends ideas from the state-of-the-art AMM classifiers (Wang et al. 2011), efficiently learning accurate, non-linear models on limited resources. Empirical evaluation was performed in a real-world ad targeting setting, using, to the best of our knowledge, the largest dataset considered thus far in the label ranking literature. The results show that the algorithm significantly outperformed the existing methods, indicating the benefits of the proposed approach to label ranking tasks.

Background

In this section we present works and ideas that led to the proposed algorithm. We first discuss label ranking setting, and then describe Adaptive Multi-hyperplane Machine (AMM), a non-linear, multi-class model used to develop a novel large-scale label ranking approach introduced in this paper.

Label ranking

Unlike standard machine learning problems such as multi-class or multi-label classification, label ranking is a relatively novel topic which involves a complex task of label preference learning. More specifically, rather than predicting one or more class labels for a newly observed example, we seek to find a strict ranking of classes by their importance or relevance to the given example. For instance, let us consider targeted advertising domain, and assume that the examples are internet users and class labels are user preferences from the set $\mathcal{Y} = \{$"sports", "travel", "finance"$\}$. Then, instead of simply inferring that the user is a "sports" person, which would result in user being shown only sports-related ads, it is more informative to know that the user prefers sports over finance over travel, resulting in more diverse and more effective ad targeting. Note that the label ranking problem differs from the learning-to-rank setup (Cao et al. 2007), where the task is to rank the examples and not labels, and can also be seen as a generalization of classification and multi-label problems (Dekel, Manning, and Singer 2003).

More formally, in the label ranking scenario the input is defined by a feature vector $\mathbf{x} \in \mathcal{X} \subset \mathcal{R}^d$, and the output is defined by a ranking $\pi \in \Pi$ of class labels. Here, the labels originate from a predefined set $\mathcal{Y} = \{1, 2, \ldots, L\}$ (e.g., $\pi = [3, 1, 4, 2]$ for $L = 4$), and Π is a set of all label permutations. Let us denote by π_i a class label at the i^{th} position in the label ranking π, and by π_i^{-1} a position (or rank) of label i in the ranking π. For instance, in the above example we would have $\pi_1 = 3$ and $\pi_1^{-1} = 2$. Then, for any i and j, where $0 \leq i < j \leq L$, we say that label π_i is *preferred* over label π_j, or equivalently $\pi_i \succ \pi_j$. Moreover, in the case of an incomplete order π, we say that any label $i \in \pi$ is preferred over the missing ones. Further, let us assume that we are given a sample from the underlying distribution $\mathcal{D} = \{(\mathbf{x}_t, \pi_t), t = 1, ..., T\}$, where π_t is a vector containing either a total or a partial order of class labels $\mathcal{Y}$. The learning goal is to find a model f that maps input examples $\mathbf{x}$ into a total ordering of labels, $f : \mathcal{X} \rightarrow \Pi$.

In the recent years the problem has seen increased attention by the machine learning community (e.g., see recent workshops and tutorials at ICML, NIPS, and other venues), and many effective algorithms have been proposed in the literature (Har-Peled, Roth, and Zimak 2003; Dekel, Manning, and Singer 2003; Kamishima and Akaho 2006; Cheng, Hühn, and Hüllermeier 2009; Grbovic, Djuric, and Vucetic 2013); for an excellent review see (Vembu and Gärtner 2011). In (Cheng, Hühn, and Hüllermeier 2009; Cheng, Dembczyński, and Hüllermeier 2010) authors propose instance-based methods for label ranking, where training examples are first clustered according to their feature vectors, and then centroid and mean ranking are found for each cluster and used for inference. This idea was extended in (Grbovic et al. 2013; Grbovic, Djuric, and Vucetic 2013), where authors use feature vectors to supervise clustering, resulting in improved performance. Apart from the prototype-based methods, often considered approaches include learning a scoring function g_i for each class, $i = 1, \ldots, L$, and sorting their output in order to infer label ranking (Elisseeff and Weston 2001; Dekel, Manning, and Singer 2003; Har-Peled, Roth, and Zimak 2003), or training a number of binary classification models to predict pairwise label preferences and aggregating their output into a total order (Hüllermeier et al. 2008; Hüllermeier and Vanderlooy 2010).

Adaptive Multi-hyperplane Machine

The AMM algorithm is a budgeted, multi-class method suitable for large-scale problems (Wang et al. 2011; Djuric et al. 2014). It is an SVM-like algorithm that formulates a non-linear model by assigning a number of linear hyperplanes to each class in order to capture data non-linearity. Given a d-dimensional example $\mathbf{x}$ and a set $\mathcal{Y}$ of L possible classes, AMM has the following form,

$$f(\mathbf{x}) = \arg\max_{i \in \mathcal{Y}} g(i, \mathbf{x}), \tag{1}$$

where the scoring function $g(i, \mathbf{x})$ for the i^{th} class,

$$g(i, \mathbf{x}) = \max_j \mathbf{w}_{i,j}^{\mathrm{T}} \mathbf{x}, \tag{2}$$

is parameterized by a weight matrix $\mathbf{W}$ written as

$$\mathbf{W} = \Big[\mathbf{w}_{1,1} \ldots \mathbf{w}_{1,b_1} | \mathbf{w}_{2,1} \ldots \mathbf{w}_{2,b_2} | \ldots | \mathbf{w}_{L,1} \ldots \mathbf{w}_{L,b_L}\Big], \tag{3}$$

where $b_1, \ldots, b_L$ are the numbers of weights (i.e., hyperplanes) assigned to each of the L classes, and each block in (3) is a set of class-specific weights. Thus, from (1) we can see that the predicted label of the example $\mathbf{x}$ is the class of the weight vector that achieves the maximum value $g(i, \mathbf{x})$.

AMM is trained by minimizing the following convex problem at each t^{th} training iteration,

$$\mathcal{L}^{(t)}(\mathbf{W}|\mathbf{z}) \equiv \frac{\lambda}{2}||\mathbf{W}||_F^2 + l(\mathbf{W}; (\mathbf{x}_t, y_t); z_t), \tag{4}$$

where λ is the regularization parameter, and the instantaneous loss $l(\cdot)$ is computed as

$$l(\mathbf{W}; (\mathbf{x}_t, y_t); z_t) = \max\left(0, 1 + \max_{i \in \mathcal{Y} \setminus y_t} g(i, \mathbf{x}_t) - \mathbf{w}_{y_t, z_t}^{\mathrm{T}} \mathbf{x}_t\right). \tag{5}$$

Element z_t of vector $\mathbf{z} = [z_1 \ldots z_T]$ determines which weight belonging to the true class of the t^{th} example is used to calculate (5), and can be fixed prior to the start of a training epoch or, as done in this paper, can be computed on-the-fly as an index of a true-class weight that provides the highest score (Wang et al. 2011).

AMM uses Stochastic Gradient Descent (SGD) to solve (4). The SGD is initialized with the zero-matrix (i.e., $\mathbf{W}^{(0)} = \mathbf{0}$), which comprises infinite number of zero-vectors for each class. This is followed by an iterative procedure, where training examples are observed one by one and

the weight matrix is modified accordingly. Upon receiving example $(\mathbf{x}_t, y_t) \in \mathcal{D}$ at the t^{th} round, $\mathbf{w}_{ij}^{(t)}$ is updated as

$$\mathbf{w}_{ij}^{(t+1)} = \mathbf{w}_{ij}^{(t)} - \eta^{(t)} \nabla_{ij}^{(t)}, \tag{6}$$

where $\eta^{(t)} = 1/(\lambda t)$ is a learning rate, and $\nabla_{ij}^{(t)}$ is the sub-gradient of (4) with respect to $\mathbf{w}_{i,j}^{(t)}$,

$$\nabla_{i,j}^{(t)} = \begin{cases} \lambda \mathbf{w}_{i,j}^{(t)} + \mathbf{x}_t, & \text{if } i = i_t, j = j_t, \\ \lambda \mathbf{w}_{i,j}^{(t)} - \mathbf{x}_t, & \text{if } i = y_t, j = z_t, \\ \lambda \mathbf{w}_{i,j}^{(t)}, & \text{otherwise,} \end{cases} \tag{7}$$

with

$$i_t = \arg\max_{k \in \mathcal{Y} \setminus y_t} g(k, \mathbf{x}) \quad \text{and} \quad j_t = \arg\max_k (\mathbf{w}_{i_t,k}^{(t)})^{\mathrm{T}} \mathbf{x}_t. \tag{8}$$

If the loss (5) at the t^{th} iteration is positive, class weight from the true class y_t indexed by z_t is moved towards $\mathbf{x}_t$ during the update, while the class weight $\mathbf{w}_{i_t,j_t}^{(t)}$ with the maximum prediction from the remaining classes is pushed away. If the updated weight is a zero-weight then it becomes non-zero, thus increasing the weight count b_i for that class by one. In this way, complexity of the model adapts to complexity of the data, and $b_i, i = 1, \ldots, L$, are learned during training.

Methodology

It has been shown that the existing label ranking methods achieve good performance on many tasks, however, in the large-scale setting considered in this paper, they might not be as effective. When faced with non-linear problems comprising millions of examples and features, the proposed methods are either too costly to train and use, or may not be expressive enough to learn complex problems. To address this issue, in this section we present a novel ranking algorithm, called AMM-rank, that extends the idea of adaptability and online learning from AMM to label ranking setting, allowing large-scale training of accurate ranking models.

AMM-rank algorithm

Before detailing the training procedure of AMM-rank, we first consider its predictive label ranking model. As discussed previously, we assume that the t^{th} training example $\mathbf{x}_t$ is associated with (possibly) incomplete label ranking π_t of length $L_t \leq L$. Given a trained AMM-rank model (3) and a test example $\mathbf{x}$, a score for each class is found using equation (2), and the predicted label ranking is obtained by sorting the scores in the descending order,

$$\hat{\pi} = \text{sort}([g(1, \mathbf{x}), g(2, \mathbf{x}), \ldots, g(L, \mathbf{x})]), \tag{9}$$

where the sort function returns indices of the sorted scores.

Training of AMM-rank resembles the training of AMM multi-class model described in the previous section. Learning is initialized with a zero-matrix comprising an infinite number of zero-vectors for each class, followed by iteratively observing examples one by one and modifying the weight matrix. At each t^{th} training iteration we minimize the following regularized instanteneous rank loss,

$$\mathcal{L}_{rank}^{(t)}(\mathbf{W}|\mathbf{z}) \equiv \frac{\lambda}{2} \|\mathbf{W}\|_F^2 + l_{rank}(\mathbf{W}; (\mathbf{x}_t, y_t); \mathbf{z}_t). \tag{10}$$

The ranking loss $l_{rank}(\cdot)$ is defined as

$$l_{rank}(\mathbf{W}; (\mathbf{x}_t, y_t); \mathbf{z}_t) = \sum_{i=1}^{L_t} \nu(i) \sum_{j=1}^{L} I(\pi_i \succ j) \max(0, 1 + g(j, \mathbf{x}_t) - \mathbf{w}_{\pi_i, z_{ti}}^{\mathrm{T}} \mathbf{x}_t), \tag{11}$$

where $\nu(i)$ is a predefined importance assigned to the i^{th} rank, and function $I(arg)$ returns 1 if arg evaluates to true, and 0 otherwise. As in label ranking setting we need to keep track of predicted scores of all L classes and not only the top one, note that we introduced vector $\mathbf{z}_t$ instead of a scalar z_t as in (4), whose element z_{ti} determines which weight belonging to label i is used to compute (10) for the t^{th} example.

Depending on the problem at hand, using the function $\nu(i)$ a modeler can emphasize the importance of some ranks over the others. For example, let us assume $\nu(i) = 1/i$. Then, in the ranking loss defined in (11), the factor i^{-1} enforces higher penalty for misranking of top-ranked topics, while the mistakes made for lower-ranked topics incur progressively smaller costs. This approach has been explored previously in information retrieval setting (Weston et al. 2012). However, it is also applicable in the context of targeted advertising, where lower-ranked classes have progressively lower relevance to an ad publisher than the higher-ranked ones. Furthermore, penalty is incurred whenever the lower-ranked label was either predicted to be preferred over the higher-ranked one, or the score of the preferred label was higher with a margin smaller than 1.

We use SGD at each training iteration to minimize the objective function (10). Subgradient of the instantaneous rank loss with respect to the weights can be computed as

$$\nabla_{i,j}^{(t)} = \lambda \mathbf{w}_{i,j}^{(t)} - \mathbf{x}_t \, I(j = z_{ti}) \, \nu(\pi_i^{-1}) \sum_{k=1}^{L} \Big(I\big(i \succ k\big) \cdot I(1 + g(k, \mathbf{x}_t) > (\mathbf{w}_{ij}^{(t)})^{\mathrm{T}} \mathbf{x}_t) \Big) + \mathbf{x}_t \, I(j = z_{ti}) \cdot \sum_{k=1}^{L} \Big(\nu(k) \, I(k \succ i) \, I\big(1 + (\mathbf{w}_{ij}^{(t)})^{\mathrm{T}} \mathbf{x}_t > (\mathbf{w}_{k z_{tk}}^{(t)})^{\mathrm{T}} \mathbf{x}_t\big) \Big). \tag{12}$$

An SGD update step (12) can be summarized as follows. At every training round all model weights are reduced towards zero by multiplying them with $(1 - 1/t)$ (the first term on the RHS). In addition, if the j^{th} weight of the i^{th} class was used to compute the score for the t^{th} label (i.e., $I(j = z_{ti})$ equals 1), it is pushed further towards $\mathbf{x}_t$ whenever the i^{th} label was either wrongly predicted to be less preferred or correctly predicted with margin smaller than 1 (the second term on the RHS). Moreover, the weight is pushed further away from $\mathbf{x}_t$ whenever the score of the class preferred over the i^{th} class was either lower or higher with margin less than 1 (the third term on the RHS). Similarly to the AMM model, the complexity of AMM-rank ranking model is automatically learned during training, and adapts to the complexity of the considered label ranking problem.

Experiments

In this section we describe the problem setting and present a large-scale, real-world data set that was used for evaluation, followed by description and analysis of empirical results.

Dataset

We are addressing a problem from display advertising domain which consists of several key players: 1) advertisers, companies that want to advertise their products; 2) publishers, websites that host the advertisements (such as Yahoo or Google); and 3) online users. The web environment provides publishers with the means to track user behavior in much greater detail than in the offline setting, including capturing user's registered information (e.g., demographics, location) and activity logs that comprise search queries, page views, email activity, ad clicks, and purchases. This brings the ability to target users based on their past behavior, which is typically referred to as ad targeting (Ahmed et al. 2011; Pandey et al. 2011; Tyler et al. 2011; Agarwal, Pandey, and Josifovski 2012; Aly et al. 2012). Having this in mind, the main motivation for the following experimental setup was the task of estimating user's ad click interests using their past activities. The idea is that, if we sort the interests in descending order of preference and attempt to predict this ranking, this task can be formulated as a label ranking problem.

The data set that was used in the empirical evaluation was generated using the information about users' online activities collected at Yahoo servers. The activities are temporal sequences of raw events that were extracted from server logs and are represented as tuples $(u_i, e_i, t_i), i = 1, \ldots, N$, where u_i is ID of a user that generated the i^{th} tuple, e_i is an event type, t_i is a timestamp, and N is a total number of recorded tuples. For each user we considered events belonging to one of the following six groups:

- page views ("pv") - website pages that the user visited;
- search queries ("sq") - user-generated search queries;
- search link clicks ("slc") - user clicks on search links;
- sponsored link clicks ("olc") - user clicks on search-advertising links that appear next to actual search links;
- ad views ("adv") - display ads that the user viewed;
- ad clicks ("adc") - display ads that the user clicked on.

Events from these six groups are all categorized into an in-house hierarchical taxonomy by an automatic categorization system and human editors. Each event is assigned to a category from a leaf of the taxonomy, and then propagated upwards toward parent categories. Considering that the server logs for each user are retained for several months, the recorded events can be used to capture users' interests in categories over long periods of time.

Following the ad categorization step, we can compute intensity and recency measures for each of L considered categories in each of the six groups. Let $\mathcal{D}_{ugct}$ denote a set of all tuples that were generated by user u, where e_i belongs to group g and is labeled with category c, with timestamp $t_i \leq t$. Then, intensity and recency are defined as follows,

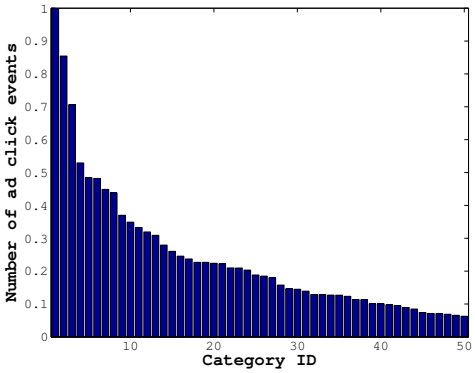

Figure 1: Number of ad click events per category

- **intensity** is an exponentially time-decayed count of all tuples in $\mathcal{D}_{ugct}$, computed as

$$intensity(u, g, c, t) = \sum_{(u_i, e_i, t_i) \in \mathcal{D}_{ugct}} \alpha^{t - t_i}, \quad (13)$$

  where α is a fixed decay factor, with $0 < \alpha < 1$ (we omit the exact value as it represents sensitive information).

- **recency** is a difference between timestamp t and a timestamp of the most recent event from $\mathcal{D}_{ugct}$, computed as

$$recency(u, g, c, t) = \min_{(u_i, e_i, t_i) \in \mathcal{D}_{ugct}} (t - t_i). \quad (14)$$

The intensity and recency measures were used to generate both the features and the label ranks for each user. In particular, we first chose two timestamps that were one month apart, $T_{features}$ and T_{labels}, where $T_{features} < T_{labels}$. Then, at timestamp $T_{features}$ we used (13) and (14) to compute intensity and recency of L categories in each of "pv", "sq", "slc", and "olc" groups separately, which, together with user's age (split into 9 buckets and represented as 9 binary features) and gender (represented as 2 binary features) was used as a feature vector $\mathbf{x}$, resulting in input space dimensionality d of $(2 \cdot 4 \cdot L + 9 + 2)$. In addition, in order to evaluate the influence of user ad views to their ad clicks, we also considered the case where intensity and recency of L categories in the "adv" group were appended to the feature vector, increasing the dimensionality by $2L$.

Furthermore, to quantify user interests and generate ground-truth ranks π, we considered only "adc" events between $T_{features}$ and T_{labels}, and computed intensity of L categories at timestamp T_{labels}. We consider level of interest of user u in category c to be equal to intensity of c in "adc" group, and preference ranking of categories is obtained simply by sorting their intensities. Note that the ground-truth ranking is in most cases incomplete, as users usually do not interact with all categories from the taxonomy.

We considered $L = 50$ second-level categories of the taxonomy (e.g., "finance/loans", "retail/apparel"), and collected data comprising 3,289,229 anonymous users that clicked on more than 2 categories. Category distribution in the ground-truth ranks is given in Fig. 1, where we see that a

Females, aged 21-25
01. Retail/Apparel
02. Technology/Internet Services
03. Telecommunications/Cellular and Wireless
04. Travel/Destinations
05. Consumer Goods/Beauty and Personal Care
06. Technology/Consumer Electronics
07. Consumer Goods/Contests and Sweepstakes
08. Travel/Vacations
09. Travel/Non US
10. Life Stages/Education

Males, aged 21-25
01. Technology/Internet Services
02. Retail/Apparel
03. Telecommunications/Cellular and Wireless
04. Travel/Destinations
05. Technology/Consumer Electronics
06. Travel/Non US
07. Travel/Vacations
08. Consumer Goods/Contests and Sweepstakes
09. Retail/Home
10. Entertainment/Games

Females, aged 65+
01. Consumer Goods/Beauty and Personal Care
02. Retail/Apparel
03. Life Stages/Education
04. Finance/Loans
05. Finance/Insurance
06. Finance/Investment
07. Technology/Internet Services
08. Entertainment/Television
09. Retail/Home
10. Telecommunications/Cellular and Wireless

Males, aged 65+
01. Finance/Investment
02. Finance/Loans
03. Retail/Apparel
04. Life Stages/Education
05. Technology/Internet Services
06. Finance/Insurance
07. Consumer Goods/Beauty and Personal Care
08. Retail/Home
09. Telecommunications/Cellular and Wireless
10. Technology/Computer Software

Figure 2: Topic ranking found by the AG-Mal model

large fraction of ad clicks would be missed if users were targeted only with the most clicked categories, which directly results in lost revenue for both publishers and advertisers.

Results

We compared AMM-rank to following approaches: a) multiclass AMM (Wang et al. 2011), where the top-ranked category was used as a true class and the output scores for all categories were sorted to obtain ranking, used as a naïve baseline; b) Central-Mal, which always predicts central ranking of the training set computed using the Mallows model (Mallows 1957); c) AG-Mal, which computes Central-Mal over all users grouped in different age ("13-17", "18-20", "21-24", "25-29", "30-34", "35-44", "45-54", "55-64", "65+") and gender (male/female) buckets; d) IB-Mal, which computes Central-Mal over k nearest neighbors (Cheng, Hühn, and Hüllermeier 2009); e) logistic regression (LR), where L binary models were trained and we sorted their outputs to obtain a ranking; and f) pairwise approach (Hüllermeier et al. 2008), where $L(L-1)/2$ binary LR models were trained and we sorted the sum of their soft votes towards each label to obtain a ranking (PW-LR). AMM-rank and PW-LR have $\mathcal{O}(NL^2)$ and IB-Mal has $\mathcal{O}(N^2L)$ time complexity, while the remaining methods are $\mathcal{O}(NL)$ approaches.

Central-Mal is a very simple and efficient baseline, and is an often-used method for basic content personalization. As the method simply predicts population's mean ranking, to improve its performance we considered AG-Mal, a method commonly used in practice, where we first compute mean rank for each age-gender group, and then use the group's mean rank as a prediction for qualified users. Further, IB-Mal is an instance-based method which is extremely competitive to the other state-of-the-art approaches (e.g., see Grbovic, Djuric, and Vucetic 2013), where we first find k nearest neighbors by considering feature vectors $\mathbf{x}$ and then predict Mallows mean ranking over the neighbors (due to large time cost, for each user we search for nearest neighbors in a subsampled set of 100,000 users). Lastly, we considered LR since it represents industry standard for ad targeting tasks, and PW-LR as it was shown to achieve state-of-the-art performance on a number of ranking tasks (Grbovic, Djuric, and Vucetic 2012b; 2013). Due to large scale of the problem, we did not consider state-of-the-art methods such as mixture models which require iterative training (Grbovic, Djuric, and Vucetic 2012a; Grbovic et al. 2013). We also did not consider log-linear model (Dekel, Manning, and Singer 2003), shown in (Grbovic, Djuric, and Vucetic 2013) to be outperformed by the IB-Mal, and do not report results of instance-based Plackett-Luce (Cheng, Dembczyński, and Hüllermeier 2010) due to observed limited performance.

We used Vowpal Wabbit package[1] for logistic regression, BudgetedSVM (Djuric et al. 2014) for AMM, that we also modified to implement AMM-rank. We set $\nu(i) = 1, i = 1, \ldots, L$, and used the default parameters from BudgetedSVM package for AMM-rank, with the exception of the λ parameter which, together with competitors' parameters, was configured through cross-validation on a small held-out set; this resulted in $k = 10$ for IB-Mal. As discussed previously, we considered two versions of the ad targeting data:

- ~~**adv**~~ - feature vector $\mathbf{x}$ does not include recency and intensity of categories from "adv" group (with $d = 411$);
- **adv**, feature vector $\mathbf{x}$ does include recency and intensity of categories from "adv" group (with $d = 511$).

Before comparing the ranking approaches, it is informative to consider the examples of label ranks found by AG-Mal on **adv** data, given in Figure 2. We can see that there exist significant differences between different gender and age groups. Albeit the obtained ranks seem very intuitive, we will see shortly that AG-Mal is significantly outperformed by the other methods, illustrating complexity of the ranking

[1] github.com/JohnLangford/vowpal_wabbit

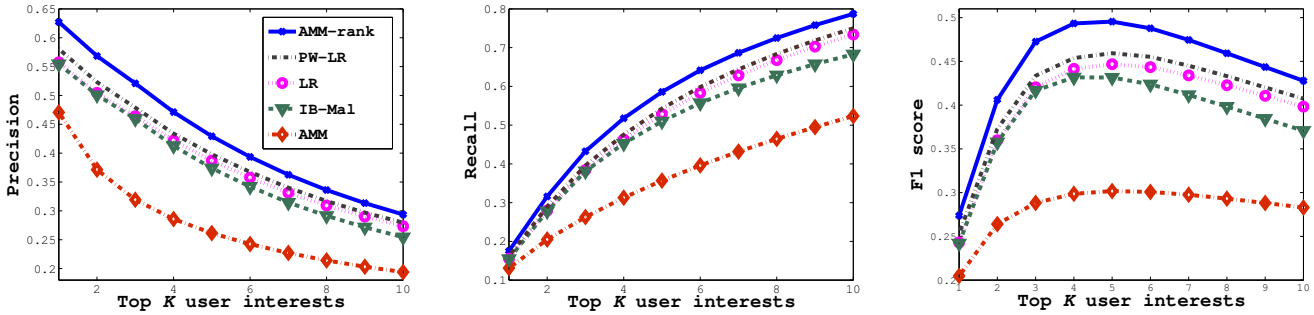

Figure 3: Comparison of retrieval performance of label ranking algorithms in terms of precision, recall, and F1 measures

Table 1: Disagreement error ϵ_{dis} of the label ranking methods

Algorithm	~~adv~~	adv
AMM	0.3446	0.2611
Central-Mal	0.2957	0.2957
AG-Mal	0.2820	0.2820
IB-Mal	0.2694	0.1899
LR	0.2110	0.1419
PW-LR	0.2091	0.1226
AMM-rank	**0.1996**	**0.1083**

task and the need for more involved approaches. In the following, we compare the algorithms using disagreement error ϵ_{dis} (Dekel, Manning, and Singer 2003), computed as a fraction of pairwise category preferences predicted incorrectly,

$$\epsilon_{\text{dis}} = \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \sum_{i,j=1}^{L} \frac{I\big((\pi_{ti} \succ \pi_{tj}) \wedge (\hat{\pi}^{-1}_{t\pi_{tj}} > \hat{\pi}^{-1}_{t\pi_{ti}})\big)}{L_t\big(L - 0.5(L_t+1)\big)}, \tag{15}$$

as well as precision, recall, and F1 at the top K ranks,

$$\begin{aligned} \text{precision@}K &= \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \sum_{i=1}^{K} \frac{I(\hat{\pi}_{ti} \in \pi_t)}{K}, \\ \text{recall@}K &= \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \sum_{i=1}^{K} \frac{I(\hat{\pi}_{ti} \in \pi_t)}{L_t}, \\ \text{F1@}K &= \frac{2 \cdot \text{precision@}K \cdot \text{recall@}K}{\text{precision@}K + \text{recall@}K}, \end{aligned} \tag{16}$$

which are commonly used measures for ranking problems. Here, $\hat{\pi}_t$ denotes predicted label rank for the t^{th} example.

Performance of the competing methods in terms of ϵ_{dis}, following 5-fold cross-validation, is reported in Table 1. We can see that the inclusion of ad view features resulted in large performance improvement, confirming findings from (Gupta et al. 2012) that past exposure to an ad increases propensity of a user to actually click the ad. As expected, multiclass AMM achieved poor performance as it optimizes only for the topmost category, and this result represents a lower bound on the disagreement loss. A simple baseline Central-Mal achieved higher error, which was decreased by only a small margin using AG-Mal. We can see that IB-Mal resulted in significant performance improvement, however in large-scale, online setting it may be very inefficient. Logistic regression, a commonly used method in ad targeting tasks, obtained low error, further improved using the pairwise approach. However, state-of-the-art PW-LR was significantly outperformed by the proposed AMM-rank which achieved more than 10% better result. We note that, other than IB-Mal, the methods are very efficient, obtaining training and test times of less than 10 minutes on a regular machine.

However, the main goal in ad targeting campaigns is not to infer the complete list of preferences for a user. Instead, we aim to find the top K most preferred categories, due to the constraint that we only have a limited budget for ad display, in terms of both time and space. Therefore, it is not of importance when two less preferred categories are misranked, and in the second set of experiments we explore how the label ranking methods perform in such setting. We considered showing $K = \{1, 2, \ldots, 10\}$ display ads, and for the top K ranks measure precision, recall, and F1 score of the categories on which the user clicked during the testing period. The results obtained by the label ranking algorithms are illustrated in Figure 3. We can see that AMM-rank outperformed the competitors, achieving better performance for all values of K. This becomes even more relevant when we consider that even a small improvement in a web-scale setting of targeted advertising may result in a significant revenue increase for the publisher. We can conclude that the results strongly suggest advantages of the proposed approach over the competing algorithms in large-scale label ranking tasks.

Conclusion

In order to address challenges brought about by the scale of the online advertising tasks that renders many state-of-the-art methods inefficient, we introduced AMM-rank, a novel, non-linear algorithm for large-scale label ranking. We evaluated its performance on a real-world ad targeting data comprising more than 3 million users, thus far the largest label ranking data considered in the literature. The results show that the method outperformed the competing approaches by a large margin in terms of both rank loss and retrieval measures, indicating that the AMM-rank algorithm is a very suitable method for solving large-scale label ranking problems.

References

Agarwal, D.; Pandey, S.; and Josifovski, V. 2012. Targeting converters for new campaigns through factor models. In *Proceedings of the 21st International Conference on World Wide Web*, 101–110. ACM.

Ahmed, A.; Low, Y.; Aly, M.; Josifovski, V.; and Smola, A. J. 2011. Scalable distributed inference of dynamic user interests for behavioral targeting. In *KDD*, 114–122.

Alba, J.; Lynch, J.; Weitz, B.; Janiszewski, C.; Lutz, R.; Sawyer, A.; and Wood, S. 1997. Interactive home shopping: consumer, retailer, and manufacturer incentives to participate in electronic marketplaces. *The Journal of Marketing* 38–53.

Aly, M.; Hatch, A.; Josifovski, V.; and Narayanan, V. K. 2012. Web-scale user modeling for targeting. In *WWW*, 3–12. ACM.

Broder, A. Z. 2008. Computational advertising and recommender systems. In *Proceedings of the ACM conference on Recommender systems*, 1–2. ACM.

Cao, Z.; Qin, T.; Liu, T.; Tsai, M.; and Li, H. 2007. Learning to rank: From pairwise approach to listwise approach. *ICML* 129–136.

Chellappa, R. K., and Sin, R. G. 2005. Personalization versus privacy: An empirical examination of the online consumers dilemma. *Information Technology and Management* 6(2-3):181–202.

Cheng, W.; Dembczyński, K.; and Hüllermeier, E. 2010. Label ranking methods based on the Plackett-Luce model. In *Proceedings of the 27th International Conference on Machine Learning*, 215–222.

Cheng, W.; Hühn, J.; and Hüllermeier, E. 2009. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th International Conference on Machine Learning*, 161–168.

Das, A. S.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: Scalable online collaborative filtering. In *WWW*, 271–280. ACM.

Dekel, O.; Manning, C.; and Singer, Y. 2003. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Djuric, N.; Lan, L.; Vucetic, S.; and Wang, Z. 2014. BudgetedSVM: A toolbox for scalable SVM approximations. *Journal of Machine Learning Research* 14:3813–3817.

Elisseeff, A., and Weston, J. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, 681–687.

Essex, D. 2009. Matchmaker, matchmaker. *Communications of the ACM* 52(5):16–17.

Grbovic, M.; Djuric, N.; Guo, S.; and Vucetic, S. 2013. Supervised clustering of label ranking data using label preference information. *Machine Learning* 1–35.

Grbovic, M.; Djuric, N.; and Vucetic, S. 2012a. Learning from pairwise preference data using Gaussian mixture model. *Preference Learning: Problems and Applications in AI* 33–35.

Grbovic, M.; Djuric, N.; and Vucetic, S. 2012b. Supervised clustering of label ranking data. In *SDM*, 94–105. SIAM.

Grbovic, M.; Djuric, N.; and Vucetic, S. 2013. Multi-prototype label ranking with novel pairwise-to-total-rank aggregation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. AAAI Press.

Gupta, N.; Das, A.; Pandey, S.; and Narayanan, V. K. 2012. Factoring past exposure in display advertising targeting. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 1204–1212.

Har-Peled, S.; Roth, D.; and Zimak, D. 2003. Constraint classification for multiclass classification and ranking. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, 785–792. MIT Press.

Hüllermeier, E., and Vanderlooy, S. 2010. Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. *Pattern Recognition* 43(1):128–142.

Hüllermeier, E.; Fürnkranz, J.; Cheng, W.; and Brinker, K. 2008. Label ranking by learning pairwise preferences. *Artificial Intelligence* 172(16):1897–1916.

Kamishima, T., and Akaho, S. 2006. Efficient clustering for orders. In *ICDM Workshops*, 274–278.

Majumder, A., and Shrivastava, N. 2013. Know your personalization: Learning topic level personalization in online services. In *Proceedings of the 22nd International Conference on World Wide Web*, 873–884.

Mallows, C. L. 1957. Non-null ranking models. *Biometrika* 44(1/2):114–130.

Manber, U.; Patel, A.; and Robison, J. 2000. Experience with personalization on Yahoo! *Communications of the ACM* 43(8):35.

Pandey, S.; Aly, M.; Bagherjeiran, A.; Hatch, A.; Ciccolo, P.; Ratnaparkhi, A.; and Zinkevich, M. 2011. Learning to target: what works for behavioral targeting. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 1805–1814. ACM.

Riecken, D. 2000. Personalized views of personalization. *Communications of the ACM* 43(8):27–28.

Tuzhilin, A. 2009. Personalization: The state of the art and future directions. *Business Computing* 3:3.

Tyler, S. K.; Pandey, S.; Gabrilovich, E.; and Josifovski, V. 2011. Retrieval models for audience selection in display advertising. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 593–598. ACM.

Vembu, S., and Gärtner, T. 2011. Label ranking algorithms: A survey. In *Preference learning*. Springer. 45–64.

Wang, Z.; Djuric, N.; Crammer, K.; and Vucetic, S. 2011. Trading representability for scalability: Adaptive multi-hyperplane machine for nonlinear classification. In *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*.

Weston, J.; Wang, C.; Weiss, R.; and Berenzweig, A. 2012. Latent collaborative retrieval. In *Proceedings of the 29th International Conference on Machine Learning*, 9–16.