Non-linear Label Ranking for Large-scale prediction of Long-Term User Interests

Nemanja Djuric[1], Vladan Radosavljevic[1], **Mihajlo Grbovic**[1], Narayan Bhamidipati[1], Slobodan Vucetic[2]

[1] Yahoo! Labs, Sunnyvale
[2] Temple University, Philadelphia

Introduction

- Ad targeting
  - Improved personalization directly translates into increased profits
  - Strategic goal of all major internet players
- For each individual user, find the ads that they are most likely to click on given their historical online behavior

- We cast the task as a label ranking problem
  - Find not only the ads that the user is likely to click on, but also sort them by the user's click propensity

Label Ranking

- We are given *d*-dimensional training points with their corresponding (possibly incomplete) rankings of *L* labels from a set $\mathcal{Y}$

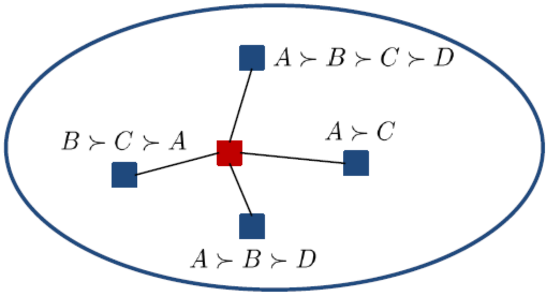
user Bob, $\mathbf{x}$ = [age, gender, browsing behavior, …]

Preference vector $\mathbf{r}$:
1. movies
2. sports
3. entertainment
4. …

- Task: Predict a ranking of labels for a new point $\mathbf{x}_{new}$
- Many proposed algorithms in the literature

Related work

- Map into classification
  - $L(L-1)/2$ classifiers, aggregate individual predictions
  - A single $(d \times L)$-dimensional problem
- *k*-NN-based algorithms
  - Aggregate ranking of *k* neighbors



- Utility functions
  - Learn score function for each label

$$f_i(\mathbf{x}) : \mathbf{x} \rightarrow R,\ i = 1, \ldots, L$$

  - Predict the ranking by sorting per-label scores

Large-scale? Non-linear??

- Existing approaches not applicable to our task:
  - Predict preferences of Yahoo users in order to improve ad targeting campaigns
  - Hundreds of millions of online users
  - Possibly highly complex mapping from input space $\mathcal{X}$ to the ranking of labels
- We propose a novel label ranking algorithm that efficiently and effectively addresses these issues

Adaptive Multi-hyperplane Machines

- Fast, large-scale, non-linear classifier
- Highly-optimized implementation available
  - BudgetedSVM, toolbox for large-scale classification
  - http://sourceforge.net/projects/budgetedsvm/
- Each class represented by a number of hyperplanes; algorithm automatically finds how many weights are actually needed according to the data complexity

AMM – Adaptive, online training

- Large-margin classifier, trained online
- Training time close to linear models, while capturing non-linearity in the data
- Model: Each class represented by b_i vectors

$$\mathbf{W} = \left[\; \mathbf{w}_{1,1} \ldots \mathbf{w}_{1,b_1} \;\middle|\; \mathbf{w}_{2,1} \ldots \mathbf{w}_{2,b_2} \;\middle|\; \ldots \;\middle|\; \mathbf{w}_{M,1} \ldots \mathbf{w}_{M,b_M} \; \right]$$

  - Prediction for the i^{th} class found as $g(i, \mathbf{x}) = \max_j \mathbf{w}_{i,j}^T \mathbf{x}$
- During training minimize the margin loss

$$\max\left(0, 1 + \max_{i \in \mathcal{Y} \setminus y_n} g(i, \mathbf{x}_n) - \mathbf{w}_{y_n, z_n}^T \mathbf{x}_n\right)$$

The proposed AMM-rank

- AMM for label ranking
  - Large-margin SVM classifiers in a new setting
  - Allows efficient and effective online training
  - Capable of capturing highly non-linear dependencies

$$loss_{rank}(\mathbf{W},(\mathbf{x}_t,\mathbf{r}_t)) = \sum_{i=1}^{|\mathbf{r}_t|} \frac{1}{i} \sum_{j=1}^{L} I(r_i > \hat{r}_j) \cdot AMM_{loss}(r_i, \hat{r}_j)$$

Higher ranks incur higher costs

Incur loss when higher and lower rank are misranked

Enforce margin between label predictions

Model training and inference

- Learn model weights using stochastic gradient descent

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

- For a test point $\mathbf{x}_{new}$ predict by sorting per-label scores

$$\hat{\pi}_{new} = \mathrm{sort}([g(1, \mathbf{x}_{new}), g(2, \mathbf{x}_{new}), \ldots, g(L, \mathbf{x}_{new})])$$

Ad targeting setting

- We considered user events: 1) ad views, 2) page views, 3) search queries, 4) search link clicks, 5) sponsored link clicks
- Each event is categorized using an in-house taxonomy
  - e.g., ‘Travel/Vacations’, ‘Finance/Loans’, ‘Sports/Football’
- Found *recency* and *intensity* for each category-event pair
  - Recency – number of days since the last event
  - Intensity – exponentially time-decayed count of all events

$$\text{recency} = \min_{i \in \text{set of all events}} (t_{current} - t_i) \qquad \text{intensity} = \sum_{i \in \text{set of all events}} \alpha^{t_{current} - t_i},\ 0 < \alpha < 1$$

Empirical evaluation

- For features $\mathbf{x}$ we used one month of user data
  - 3,289,229 users, we considered events categorized into 50 most frequent second-level categories of the taxonomy
  - Computed recency and intensity of the 50 categories for each of the 5 user events, and used 9 age and 2 gender indicators
  - Resulted in (2 x 5 x 50 + 9 + 2) = 511-dimensional input space
- To generate label ranking $\mathbf{r}$ for a user, we sorted intensity of categorized ad clicks in the following two-weeks period

Baseline methods

1. AMM-rank: Multi-class method used on label ranking
2. Central-Mal: Predict a single global Mallows ranking
3. AG-Mal: Central-Mal over all age-gender buckets
   - Age groups: 13-17, 18-20, 21-24, 25-29, 30-34, 35-44, 45-54, 55-64, 65+
4. IB-Mal: Central-Mal over k-nearest neighbors (k=10)
5. Logistic Regression (LR): Train L separate LR methods
6. PW-LR: Train $L(L-1)/2$ pairwise LR models

Example

- Ranking of 50 taxonomy categories using AG-Mal

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

Females, aged 65+
01. Consumer Goods/Beauty & 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 & Wireless

Results

- We report label disagreement loss
  - Percentage of pairs of misranked labels

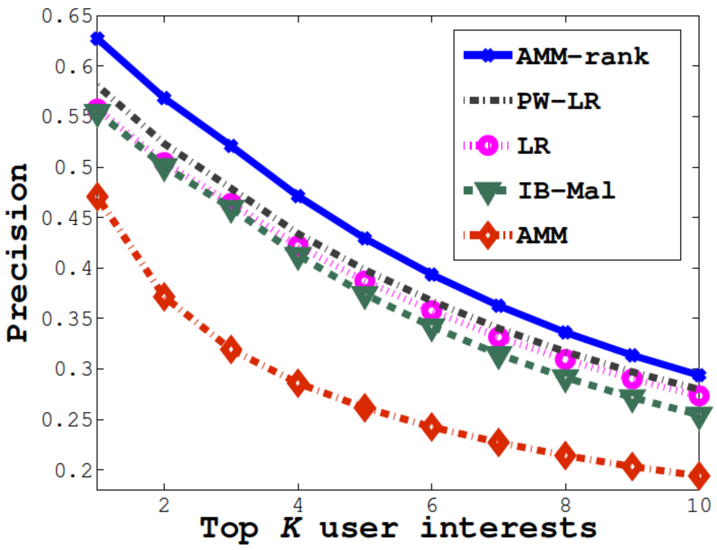
$$\epsilon_{\text{dis}} = \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \sum_{i,j=1}^{L} \frac{I(\pi_{ti} \succ \pi_{tj} \wedge \hat{\pi}^{-1}_{t\pi_{tj}} > \hat{\pi}^{-1}_{t\pi_{ti}})}{L_t\big(L - 0.5(L_t + 1)\big)}$$

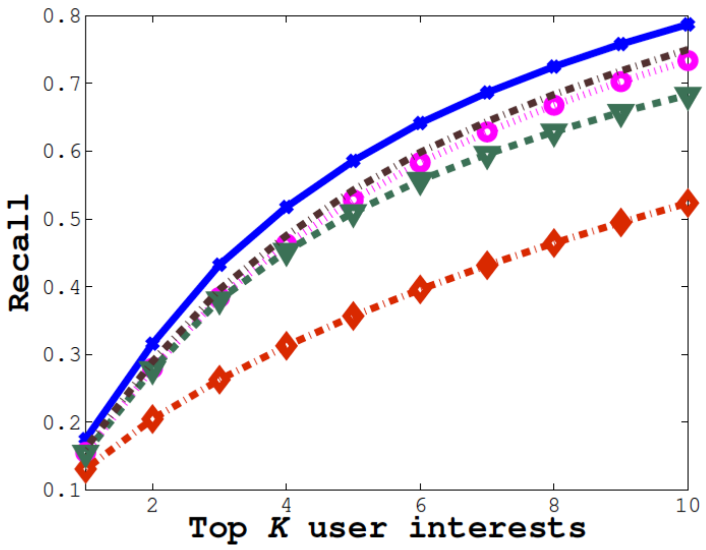
- Computed the loss using data with and without ad views
  - Ad views carry a strong signal, although not user actions

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**

Results

- Precision and recall in the top *K* interests
  - AMM-rank significantly outperforms the competing methods





Conclusion

- The proposed AMM-rank learns **non-linear** mapping between users and label ranking
- State-of-the-art performance on **limited memory**
- Training on 3.3 million Yahoo users runs in less than 10 minutes, **outperforming** the competing methods
- **Highly efficient** algorithm for label ranking

Thank you!

- Questions and/or suggestions?