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

Nemanja Djuric¹, Vladan Radosavljevic¹, Mihajlo Grbovic¹, Narayan Bhamidipati¹, Slobodan Vucetic²

¹ Yahoo! Labs, Sunnyvale
² Temple University, Philadelphia
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, $\mathbf{x} = [\text{age, gender, browsing behavior, ...}]$

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

- Task: Predict a ranking of labels for a new point $\mathbf{x}_{\text{new}}$

- 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
    $$f_i(x): x \rightarrow R, \ i = 1, \ldots, L$$
  - 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/](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

$$W = \begin{bmatrix} w_{1,1} \ldots w_{1,b_1} & w_{2,1} \ldots w_{2,b_2} & \ldots & w_{M,1} \ldots w_{M,b_M} \end{bmatrix}$$

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

$$\max \left( 0, 1 + \max_{i \in \mathcal{Y} \setminus y_n} g(i, x_n) - w_{y_n, z_n}^T x_n \right)$$
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

\[
\text{loss}_{\text{rank}}(\mathbf{W}, (\mathbf{x}_t, r_t)) = \sum_{i=1}^{\left| r_t \right|} \frac{1}{L} \sum_{j=1}^L I(r_i > \hat{r}_j) \cdot \text{AMM}_{\text{loss}}(r_i, \hat{r}_j)
\]

- 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

\[ \nabla_{i,j}^{(t)} = \lambda w_{i,j}^{(t)} - x_t I(j = z_{ti}) \nu(\pi_i^{-1}) \sum_{k=1}^{L} (I(i > k) \cdot I(1 + g(k, x_t) > w_{ij}^{(t)} x_t)) \\
+ x_t I(j = z_{ti}) \cdot \sum_{k=1}^{L} (\nu(k) I(k > i) I(1 + w_{ij}^{(t)} x_t > w_{kztk}^{(t)} x_t)) \]

- For a test point \( x_{new} \) predict by sorting per-label scores

\[ \hat{\pi}_{new} = \text{sort}([g(1, x_{new}), g(2, x_{new}), \ldots, g(L, x_{new})]) \]
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

\[
\text{recency} = \min_{i \in \text{set of all events}} (t_{current} - t_i) \quad \text{intensity} = \sum_{i \in \text{set of all events}} \alpha^{t_{current} - t_i}, \quad 0 < \alpha < 1
\]
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 $(2 \times 5 \times 50 + 9 + 2) = 511$-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

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

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>adv</th>
<th>adv</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMM</td>
<td>0.3446</td>
<td>0.2611</td>
</tr>
<tr>
<td>Central-Mal</td>
<td>0.2957</td>
<td>0.2957</td>
</tr>
<tr>
<td>AG-Mal</td>
<td>0.2820</td>
<td>0.2820</td>
</tr>
<tr>
<td>IB-Mal</td>
<td>0.2694</td>
<td>0.1899</td>
</tr>
<tr>
<td>LR</td>
<td>0.2110</td>
<td>0.1419</td>
</tr>
<tr>
<td>PW-LR</td>
<td>0.2091</td>
<td>0.1226</td>
</tr>
<tr>
<td>AMM-rank</td>
<td>0.1996</td>
<td>0.1083</td>
</tr>
</tbody>
</table>
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?