Online Ranking Combination

Erzsébet Frigó

Institute for Computer Science and Control (MTA SZTAKI)

Joint work with Levente Kocsis
Overview

- Framework: prequential ranking evaluation
- Goal: optimize convex combination of ranking models
- Our proposal: direct optimization of the ranking function
Objective: choosing combination weights.
Model combination in prequential framework with ranking evaluation

Objective: choosing combination weights.
New idea: optimize ranking function directly

- Standard method: take a surrogate function and use its gradient
  - E.g. MSE
- Drawback: optimum of the surrogate ≠ optimum of ranking function
- Proposed solution: optimize the ranking function directly
- Two approaches:
  - Global search in the weight space
  - Gradient approximation (finite differences)
Choose a subset $Q$ of the weight space $\Theta$
- e.g., lay a grid to the parameter space
- Apply exponentially weighted forecaster on $Q$

$$P(\text{select } q \in Q \text{ in round } t) = \frac{e^{-\eta t \sum_{\tau=1}^{t-1}(1-r_\tau(q))}}{\sum_{s \in Q} e^{-\eta t \sum_{\tau=1}^{t-1}(1-r_\tau(s))}}$$

Theoretical guarantee:

$$E[R_T(\text{best static combination in } \Theta) - R_T(\text{ExpW})] < O(\sqrt{T})$$
- if the cumulative reward function ($R_T$) is sufficiently smooth
- and $Q$ is sufficiently large

Difficulty: size of $Q$ is exponential in number of base rankers, can't scale
Approximated gradient (for the weight of base ranker $i$ in round $t$):

$$g_{ti} = \frac{r_t(\theta_t + c_t \Delta_t) - r_t(\theta_t - c_t \Delta_t)}{c_t \Delta_{ti}}$$

- $\theta_t$ is the current combination weight vector
- $\Delta_t = (\Delta_{t1}, \ldots)$ is a random vector of +/-1
- $c_t$ is perturbation step size

Online update step: one gradient step using the approximated gradient
RSPSA

- RSPSA = SPSA + Resilient Backpropagation (RProp)
RSPSA = SPSA + Resilient Backpropagation (RProp)

- RProp defines gradient step sizes for each weight
- Perturbation step size is tied to gradient step size
- Update step sizes using RProp
Resilient Backpropagation (RProp)

- Gradient update rule
- Predefined step size for each coordinate
  - ignores the length of the gradient vector
- Step size is updated based on the sign of the gradient
  - decrease step if gradient changed direction
  - increase otherwise
\[ g_{ti} = \frac{r_t(\theta_t + c_t \Delta_t) - r_t(\theta_t - c_t \Delta_t)}{c_t \Delta_{ti}} \]
**RFDSA+**

- Switch to finite differences (FD)
  - allows to detect 0 gradient w.r.t. one coordinate
- If the gradient is 0 w.r.t. a coordinate, then
  - increase perturbation size (+) for that coordinate
  - escape flat section in the right direction
- RFDSA+ = RSPSA - simultaneous perturbation + finite differences + zero gradient detection
- The modifications might seem to be minor, but are essential to make the algorithm work
Experiments - Datasets, base rankers

- 5 datasets
  - Amazon
    - CDs and Vinyl
    - Movies and TV
    - Electronics
  - MovieLens 10M
  - Twitter
    - hashtag prediction
- Size
  - # of events: 2M-10M
  - # of users: 70k-4M
  - # of items: 10k-100k

Base rankers:
- Models updated incrementally
  - SGD Matrix Factorization
  - Asymmetric Matrix Factorization
  - Item-to-item similarity
  - Most popular
- Traditional models updated periodically
  - SGD Matrix Factorization
  - Implicit Alternating Least Squares MF
Combination algorithms in the experiments

Direct optimization:
- **ExpW**
  - exponentially weighted forecaster on a grid
  - global optimization
- **SPSA**
  - gradient method with simultaneous perturbation
- **RSPSA**
  - SPSA with RProp
- **RFDSA+**
  - our new algorithm
  - finite differences, flat section detection

Baselines:
- **ExpA**
  - exponentially weighted forecaster on the base rankers
- **ExpAW**
  - use probabilities of ExpA as weights
- **SGD**
  - use MSE as a surrogate
  - target=1 for positive sample
  - target=0 for generated negative samples
Results - 2 base rankers - Combination weights

OptG100+
ExpAW
SGD
SPSA
RSPSA
RFDSA+
Cumulative reward as function of combination weight

\[ R_T(\theta) \]
Results - Scalability

NDCG vs number of OMF's for different algorithms:

- ExpA
- ExpAW
- SGD
- SPSA
- RSPSA
- RFDSA+

Graph shows the performance (NDCG) for varying numbers of OMFs.
## Results - 6 base rankers - DCG

<table>
<thead>
<tr>
<th></th>
<th>item2item</th>
<th>OMF</th>
<th>Pop</th>
<th>OAMF</th>
<th>MF</th>
<th>iALS</th>
<th>ExpA</th>
<th>ExpAW</th>
<th>SGD</th>
<th>SPSA</th>
<th>RSPSA</th>
<th>RFDSA+</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CD</strong></td>
<td>0.03</td>
<td>0.04</td>
<td>0.06</td>
<td>0.03</td>
<td>0.01</td>
<td>0.00</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.07</td>
<td>0.06</td>
<td>0.09</td>
</tr>
<tr>
<td><strong>Amazon</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Movies</td>
<td>0.04</td>
<td>0.04</td>
<td>0.07</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>Electro</td>
<td>0.02</td>
<td>0.02</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
<td>0.04</td>
<td>0.05</td>
</tr>
<tr>
<td>MovieLens</td>
<td>0.14</td>
<td>0.14</td>
<td>0.09</td>
<td>0.17</td>
<td>0.01</td>
<td>0.01</td>
<td>0.17</td>
<td>0.17</td>
<td>0.16</td>
<td>0.17</td>
<td>0.14</td>
<td>0.19</td>
</tr>
<tr>
<td>Twitter</td>
<td>0.02</td>
<td>0.35</td>
<td>0.35</td>
<td>0.31</td>
<td>0.01</td>
<td>0.01</td>
<td>0.35</td>
<td>0.35</td>
<td>0.36</td>
<td>0.37</td>
<td>0.45</td>
<td>0.46</td>
</tr>
</tbody>
</table>
Conclusions

- Problem: combine ranking algorithms
- Our proposal: optimize the ranking measure directly
- Global optimization (ExpW) works well in case of two base algo
- Our new algo: RFDSA+
  - solves problems (scaling, constant sections w.r.t one coordinate)
  - strong combination
Online Ranking Combination

Erzsébet Frigó

Institute for Computer Science and Control (MTA SZTAKI)

Joint work with Levente Kocsis