PRanking with Ranking

Koby Crammer
Technion – Israel Institute of Technology

Based on joint work with Yoram Singer at the Hebrew University of Jerusalem
Problem Setting

- Instances \( x \in \chi \)
- Labels \( y \in \{1,2,\ldots,k\} \)
- Structure \( 1 < 2 < 3 < 4 < 5 \)
- Ranking rule \( f : \chi \rightarrow \{1,2,\ldots,k\} \)

Online Framework

- Algorithm works in rounds
- On each round the ranking algorithm:
  - Gets an input instance
  - Outputs a rank as prediction
  - Receives the correct rank-value
  - Computes loss
  - Updates the rank-prediction rule

\[ x_1 \text{ is preferred over } x_2 \]

\[ f(x_1) \succ f(x_2) \]
Goal

• Algorithms Loss
  \[ L_t = \sum_{i=1}^{\dagger} |y_i - \hat{y}_i| \]

• Loss of a fixed function
  \[ L_t(f) = \sum_{i=1}^{\dagger} |y_i - f(x_i)| \]

• Regret
  \[ L_t - \inf_{f \in F} L_t(f) \]

• No statistical assumptions over data
• The algorithm should do well irrespectively of specific sequence of inputs and target labels
Background

Binary Classification
The Perceptron Algorithm
Rosenblatt, 1958

- Hyperplane $\mathbf{w}$
The Perceptron Algorithm

Rosenblatt, 1958

- Hyperplane $w$
- Get new instance $x$
- Classify $x$:
  
  $\text{sign}(w \cdot x)$
The Perceptron Algorithm
Rosenblatt, 1958

- Hyperplane \( \mathbf{w} \)
- Get new instance \( \mathbf{x} \)
- Classify \( \mathbf{x} \):
  \[
  \text{sign}(\mathbf{w} \cdot \mathbf{x})
  \]
- Update (in case of a mistake)
  \[
  \mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}
  \]
A Function Class for Ranking
Our Approach to Ranking

- Project

\[ x \rightarrow w \cdot x \]
Our Approach to Ranking

- Project
  \[ x \rightarrow w \cdot x \]
- Apply Thresholds
  \[ w \cdot x \leq b_r \]

Diagram:
- Points are projected along the vector \( w \).
- The threshold values are indicated as \( b_1, b_2, b_3 \).
- The rank is determined by the position relative to the thresholds.

\[ \text{Rank} \]
Update of a Specific Algorithm

FIX it if it's not Broken
Least change as possible
One step at a time
PRank

- Direction $\mathbf{w}$,
- Thresholds $b_{k-1}, \ldots, b_1$
PRank

- Direction $\mathbf{w}$
- Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance $\mathbf{x}$
PRank

Correct Rank Interval

- Direction $w$, Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance $x$
- Get the correct rank $y$
PRank

- Direction \( w \), Thresholds \( b_{k-1}, \ldots, b_1 \)
- Rank a new instance \( x \)
- Get the correct rank \( y \)
- Compute Error-Set \( E \)

\[ E = \{ b_2, b_3 \} \]

\( b_1 \) \( b_2 \) \( b_3 \) \( b_4 \)

1 2 3 4 5
PRank – Update

\[ E = \{ b_2, b_3 \} \]

- Direction \( w \),
- Thresholds \( b_{k-1}, \ldots, b_1 \)
- Rank a new instance \( x \)
- Get the correct rank \( y \)
- Compute Error-Set \( E \)
- Update:

\[ b_r \leftarrow b_r - 1 \quad \forall r \in E \]
$E = \{ b_2, b_3 \}$

- Direction $w$
- Thresholds $b_{k-1}, \ldots, b_1$
- Rank a new instance $x$
- Get the correct rank $y$
- Compute Error-Set $E$
- Update:
  - $b_r \leftarrow b_r - 1 \quad r \in E$
  - $w \leftarrow w + |E| x$
PRank – Summary of Update

\[ \mathcal{E} = \left\{ b_2, b_3 \right\} \]

- Direction \( w \),
- Thresholds \( b_{k-1}, \ldots, b_1 \)
- Rank a new instance \( x \)
- Get the correct rank \( y \)
- Compute Error-Set \( \mathcal{E} \)
- Update:
  
  1. \( b_r \leftarrow b_r - 1 \) \( r \in \mathcal{E} \)
  2. \( w \leftarrow w + |\mathcal{E}|x \)
The PRank Algorithm

Maintain $w, b_1, \ldots, b_{k-1}$

Get an instance $x$

Predict: $\hat{y} = 1 + \# \{ r : w \cdot x < b_r \}$

Get the true rank $y$

Compute Error set: $E = \{ r : y \leq r < \hat{y} \}$

Update

$w \leftarrow w + |E|x$

$b_r \leftarrow b_r - 1 \quad r \in E$

Yes $

E \neq \Phi$

No
Analysis

Two Lemmas
Consistency

• Can the following happen?
Can the following happen? No

The order of the thresholds is preserved after each round of PRank: $b_1 \leq \ldots \leq b_{k-1}$
Regret Bound

Given:
- Arbitrary input sequence

Easy Case:
- Assume there exists a model that ranks all the input instances correctly
  - The total loss the algorithm suffers is bounded

Hard Case:
- In general
  - A “regret” $L_t - \inf_{f \in F} \tilde{L}_t(f)$ is bounded
Ranking Margin

\[ \text{Margin}(x,y) = \min \left\{ \min_{r \geq y} \{ w \cdot x - b_r \}, \min_{r < y} \{ b_r - w \cdot x \} \right\} \]
Ranking Margin

\[ \text{Margin}(x,y) = \min \left\{ \min_{r \geq y} \{w \cdot x - b_r\}, \min_{r < y} \{b_r - w \cdot x\} \right\} \]
Margin(x,y) = \min \left\{ \min_{r \geq y} \{w \cdot x - b_r\}, \min_{r < y} \{b_r - w \cdot x\} \right\}
Ranking Margin

\[
\text{Margin}(x, y) = \min \left\{ \min_{r \geq y} \{ w \cdot x - b_r \}, \min_{r < y} \{ b_r - w \cdot x \} \right\}
\]

\[
\text{Margin}\left( (x^1, y^1), \ldots, (x^T, y^T) \right) = \min \text{ Margin}\left( x^+, y^+ \right)
\]
Mistake Bound

Given:

- Input sequence \((x^1, y^1), ..., (x^T, y^T)\),
- Norm of instances is bounded \(\|x^t\|^2 \leq R^2\)
- Ranked correctly by a normalized ranker
  \(\|w\|^2 + b_1^2 + ... + b_{k-1}^2 = 1\) with Margin\(>0\)

Then:

\[
\text{Number of Mistakes PRank Makes} \leq (k - 1) \frac{R^2 + 1}{\text{Margin}^2}
\]
## Exploit Structure

<table>
<thead>
<tr>
<th></th>
<th>Loss</th>
<th>Range</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>$y \neq \hat{y}$</td>
<td>$y \in {1, 2, \ldots, k}$</td>
<td>None</td>
</tr>
<tr>
<td>Regression</td>
<td>$(y - \hat{y})^2$</td>
<td>$y \in \mathbb{R}$</td>
<td>Metric</td>
</tr>
<tr>
<td>Ranking</td>
<td>$</td>
<td>y - \hat{y}</td>
<td>$</td>
</tr>
</tbody>
</table>

- Under Constraint
- Over Constraint
Other Approaches

• Treat Ranking as Classification or Regression
  E.g. Basu, Hirsh, Cohen 1998

• Reduce a ranking problem into a classification problem over pair of examples
  E.g. Freund, Lyer, Schapire, Singer 1998
  Herbrich, Graepel, Obermayer 2000
  – Not simple to combine preferences predictions over pairs into a single consistent ordering
  – No simple adaptation for online settings
Empirical Study
An Illustration

- Five concentric ellipses
- Training set of 50 points
- Three approaches
  - Pranking
  - Classification
  - Regression

PRank
Ranking

MC-Perceptron
Classification

Widrow-Hoff
Regression
Each-Movie database

- 74424 registered Users
- 1648 listed Movies
- Users ranking of movies
- 7451 Users saw >100 movies
- 1801 Users saw >200 movies
Ranking Loss, 100 Viewers

- Regression
- Classification
- PRank

Round

Over constrained
Under constrained
Accurately constrained
Ranking Loss, 200 Viewers

![Graph showing ranking loss over rounds for Regression, Classification, and PRank.]
AT THE END OF A LONG CODING DAY
OR
THE CANARY IN THE CODE MINE

MAYBE WE SHOULD OPEN A WINDOW

REAL GEEKS DON'T NEED OXYGENE
(1) User choose movies from this list

(2) Movies chosen and ranked by user
(3) Press the ‘learn’ key. The system learns the user’s taste.

(4) The system re-ranks the training set.

(5) The system re-ranks a new fresh set of yet unseen movies.
(6) Press the ‘flip’ button to see what movies you should not view

(7) The flipped list
Many alternatives to formulate ranking
Choose one that models best your problem
Exploit and Incorporate structure
Specifically:
- Online algorithm for ranking problems via projections and conservative update of the projection’s direction and the threshold values
- Experiments on a synthetic dataset and on Each-Movie data set indicate that the PRank algorithm performs better than algorithms for classification and regression