Ranking with a P-Norm Push

Cynthia Rudin

NSF Postdoctoral Research Fellow
NYU Center for Neural Science and Courant Institute

Impromptu session, Snowbird 2006
Movies that I like:

- Secret Lives of Dentists
- The Incredibles
- Big Fish
- Hitch
- Grizzly Man
- Amelie
- UHF
- Catch Me If You Can

Movies that I don’t:

- Titanic
- Jurassic Park
- Howard the Duck
- True Lies
- Finding Nemo
- True Lies
- Finding Nemo
Consider the following real-world problem:

Cynthia wants to go to the movies on Friday night…
… and she would like to see a good movie.

Where can she find a good movie recommendation given her taste in movies?
<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>Worldwide Box Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Titanic (1997)</td>
<td>$1,835,300,000</td>
</tr>
<tr>
<td>3.</td>
<td>Harry Potter and the Sorcerer's Stone (2001)</td>
<td>$968,600,000</td>
</tr>
<tr>
<td>5.</td>
<td>The Lord of the Rings: The Two Towers (2002)</td>
<td>$921,600,000</td>
</tr>
<tr>
<td>6.</td>
<td>Jurassic Park (1993)</td>
<td>$919,700,000</td>
</tr>
<tr>
<td>8.</td>
<td>Harry Potter and the Chamber of Secrets (2002)</td>
<td>$866,300,000</td>
</tr>
<tr>
<td>9.</td>
<td>Finding Nemo (2003)</td>
<td>$865,000,000</td>
</tr>
</tbody>
</table>
• I don’t want a combined list that’s supposed to work for everybody.

• I want personalized rankings, so it’s a supervised learning problem.

• Remember, the best movies should be at the top!

   (This is where the p-norms are going to come in handy!)
The Problem of Supervised Bipartite Ranking

Given: *Training Data*

\[ x^+_i \in X, \; i=1..I, \; \text{chosen iid from unknown probability distribution} \; D_+. \]
Also \( x^-_k \in X, \; k=1..K, \; \text{chosen from} \; D_- . \)
The Problem of Supervised Bipartite Ranking

Given: *Training Data*

\[ x^+_i \in X, \ i = 1..I, \ \text{chosen iid from unknown probability distribution } D_+ \]

Also \( x^-_k \in X, \ k = 1..K, \ \text{chosen from } D_- \).

Our Goal: Construct a function \( f : X \rightarrow \mathbb{R} \) such that for \( x_+ \sim D_+ \) and \( x_- \sim D_- \), we have \( f(x_+) > f(x_-) \) with high probability.

(Notation: \( x \sim D \) means \( x \) chosen randomly from \( D \).)

But…
- Waterworld
+ Amelie
+ Big Fish
+
+ The top of the list is most important!
-
+
+ If this is our ranked list...
-
-
-
-
-
-
+
- True Lies
+ Hitch
- Titanic
By the way, this ranking problem isn’t just for movies:

- meta search for search engines
- fraud detection (900 phone numbers, sponsored links)
- processes to be completed in a certain order
- natural language processing
- bioinformatics
- pharmaseudicals

HUGE number of applications
Outline of Talk

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

1) Deriving an Objective Function
2) The “P-Norm Push” Algorithm
3) A Generalization Bound
4) Uniqueness

(Rudin, COLT 2006)
How to measure the goodness of a ranked list? use ROC curves!

- Algorithms for ranking often evaluate the AUC (Area Under the ROC)
AUC Optimization

• Algorithms for ranking often evaluate the AUC (Area Under the ROC)

• Boosting AUC Maximizers:
  - RankBoost (Freund et al.)
  - AdaBoost (Freund & Schapire '97) (see Rudin et al. '05)

• SVM AUC Maximizers: (Yan et al. '03), (Rakotomamonjy '04)

• Other AUC Maximizers: (Fawcett '01), (Bostrom '05), (Liu and Wu),
  (Zhang et al. '02), (Ferri et al. '03), …

• Tutorials on AUC:
  • (Flach '04) At ICML 2004
  • J. A. Hanley and B. J. McNeil. The meaning and use of the area under the receiver operating characteristic (roc) curve. Radiology, 143:29-36, 1982
How to measure the goodness of a ranked list? use ROC curves!
How to measure the goodness of a ranked list? use ROC curves!

• Algorithms for ranking often evaluate the AUC (Area Under the ROC)
  • The AUC concentrates uniformly along the ranked list

• Our problem is slightly different!
  • We care mostly about the leftmost portion of the ROC curve
  • It’s ok to sacrifice the rightmost portion a bit.
  • Only a small amount of literature on this problem:
    (Mozer et al. 02, Yan et al. 03)

  Other related literature: “Log-Linear Models for Label Ranking” (Dekel et al. 03), “Permutation Groups” (Lebanon Lafferty 02), “Partial AUC” (Dodd&Pepe 03)
Deriving A Convex Objective

\[ \text{Height}_{\text{of}} (k) = \sum_{i=1}^{l} \left[ f(x_i^+) \leq f(x_k^-) \right] = \text{Number of positive examples ranked below negative example } k \]

If \( k \) is ranked above \( k_1 \) then:

\[ \text{Height}_{\text{of}} (k) = \text{Height}_{\text{of}} (k_1) \]

We want to concentrate harder on \( k \) than on \( k_1 \).
• Put a price on each negative example:

Price for example $k$ is:

$$g \left( \sum_{i=1}^{I} 1 - f(x_i^+) \leq f(x_k^-) \right)$$

where $g: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is convex, monotonically increasing.

Note: When $g(z) = z$, then $R_{g,1} = \text{const} \cdot (1 - \text{AUC})$.

• For example,

$$g(z) = z, \quad g(z) = \exp(z), \quad g(z) = z^p \text{ for } p \text{ large}$$

• Objective function

$$R_{g,1}(f) := \sum_{k=1}^{K} g \left( \sum_{i=1}^{I} 1 - f(x_i^+) \leq f(x_k^-) \right)$$
Let $g(z) = z^4$

Original Total Price: $2^4 + 2^4 + 1^4 = 33$

Swap lower pair: $2^4 + 2^4 + 1^4 + 1^4 = 34$

Swap higher pair: $3^4 + 2^4 + 1^4 = 98$

$$R_{g,1}(f) := \sum_{k=1}^{K} g \left( \sum_{i=1}^{I} 1 \left( f(x_i^+) \leq f(x_k^-) \right) \right)$$
0-1 Objective:

\[
R_{g,1}(f) := \sum_{k=1}^{K} g \left( \sum_{i=1}^{I} 1_{f(x_i^+) \leq f(x_k^-)} \right)
\]

Related convex objective:

\[
R_{g,\ell}(f) := \sum_{k=1}^{K} g \left( \sum_{i=1}^{I} \ell(f(x_i^+) - f(x_k^-)) \right)
\]

where \( \ell: \mathbb{R} \to \mathbb{R}_+ \) is a convex upper bound on \( 1_{z \leq 0} \).
The “P-Norm Push” Algorithm

Related convex objective:

\[
R_{g,\ell}(f) := \sum_{k=1}^{K} g\left( \sum_{i=1}^{I} \ell(f(x_i^+) - f(x_k^-)) \right)
\]

Choose \( \ell(z) := \exp(-z) \)

\[
g(z) := z^p \text{ for } p \text{ large}
\]

\[
F_p(f) := \sum_{k=1}^{K} \left( \sum_{i=1}^{I} \exp(-f(x_i^+) + f(x_k^-)) \right)^p
\]

Choose a form for \( f \) (i.e., choose an appropriate hypothesis space)

Boosting-type approach:

\[
f(x) = \sum_{j=1}^{n} \lambda_j h_j(x)
\]

Where \( h_j : X \rightarrow [0,1] \) are “weak rankers”. 
The “P-Norm Push” Algorithm

Related convex objective:

\[ R_{g,\ell}(f) := \sum_{k=1}^{K} g\left( \sum_{i=1}^{I} \ell(f(x_i^+) - f(x_k^-)) \right) \]

Choose \( \ell(z) := \exp(-z) \)

\( g(z) := z^p \) for \( p \) large

Boosting algorithms combine weak learning rules to create a strong learning rule. (Schapire ‘89)

\( h_1 = \text{Movies where most of the characters survive to the end are good} \)

\( h_1(\text{Amelie})=1, h_1(\text{Spiderman})=1, \)
\( h_1(\text{The Matrix})=0, h_1(\text{Titanic})=0, h_1(\text{Boogeyman})=0 \)

\( h_n = \text{Classic superhero movies are good} \)

\( h_n(\text{Spiderman})=1, \)
\( h_n(\text{The Matrix})=0, h_n(\text{Titanic})=0, h_n(\text{Boogeyman})=0, h_n(\text{Amelie})=0 \)

Where \( h_j : X \rightarrow [0,1] \) are “weak rankers”.
The “P-Norm Push” Algorithm

\[
\min_{\lambda \in \mathbb{R}^n} F_p (\lambda) := \sum_{k=1}^{K} \left( \sum_{i=1}^{I} \exp(-f(x_i^+) + f(x_k^-)) \right)^p
\]

where \( f(x) = \sum_{j=1}^{n} \lambda_j h_j(x) \) and \( h_j : X \rightarrow [0,1] \) \( j = 1, \ldots, n \)

• minimization is convex!
• use coordinate descent to optimize
• algorithm is pretty simple
• generalizes RankBoost (take \( p=1 \)).
The “P-Norm Push” Algorithm

**Input**: \{x_i^+\}_{i=1,...,I} positive examples, \{x_k^-\}_{k=1,...,K} negative examples, 
{h_j}_{j=1,...,n} weak rankers, \(t_{\text{max}}\) number of iterations, \(p\) power

**Initialize**: \(\lambda_{1,j} = 0\) for \(j = 1,...,n\), \(d_{1,ik} = 1 / IK\) for \(i = 1,...,I, k = 1,...,K\), 
\(M_{ikj} = h_j(x_i^+) - h_j(x_k^-)\) for all \(i, k, j\)

**Loop for** \(t = 1,...,t_{\text{max}}\)

(a) \(j_t \in \arg\max_j \left[ \sum_{k=1}^{K} \left( \sum_{i=1}^{I} d_{t,ik} \right)^{p-1} \sum_{i=1}^{I} d_{t,ik} M_{ikj} \right] \)

(b) Perform a linesearch for \(\alpha_t\) (details omitted)

(c) \(\lambda_{t+1} = \lambda_t + \alpha_t e_{j_t}\) where \(e_{j_t}\) is 1 in position \(j_t\) and 0 elsewhere.

(d) \(d_{t+1,ik} = d_{t,ik} \exp[(-M\lambda_t)_{ik}] / z_t\) for \(i = 1,...,I, k = 1,...,K\)

where \(z_t\) is a normalization factor.

**Output**: \(\lambda_{t_{\text{max}}}\)
The “P-Norm Push” Algorithm

Experimentally, what do we expect?

```
<table>
<thead>
<tr>
<th></th>
<th>p=1</th>
<th>p=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>“true positives”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“false positives”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
UCI Data - Tic Tac Toe - Training

Tic-Tac-Toe - Training

True Positives vs False Positives for different values of $p$: $p=1$, $p=2$, $p=4$, $p=8$, $p=16$, $p=64$.

Tic-Tac-Toe - Training

True Positives vs False Positives for different ranges of $p$. The graph shows a clear trend with increasing $p$.
UCI Data – Pima Indians Diabetes Threshold - Training

Pima-Indians-Diabetes with Threshold Features - Training

True Positives

False Positives

p=64
p=16
p=8
p=4
p=2
p=1

Pima-Indians-Diabetes with Threshold Features - Training

True Positives

False Positives

0 2 4 6 8

0 50 100 150
UCI Data - Boston Housing - **Testing**

**Housing - Testing**

![Graph showing True Positives and False Positives with different values of p (1, 2, 4, 8, 16, 64).](image)
Using P-Norm Ranking Algorithm for NLP

• Joint with Heng Ji and Ralph Grishman (NYU)

• Chinese Name Tagging

• Weighted crucial pairs formulation of P-Norm Objective

Outline of Talk

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

1) Deriving an Objective Function

2) The “P-Norm Push” Algorithm

3) A Generalization Bound

4) Uniqueness

A convex objective exists which concentrates near the top

Performs well on UCI data. Easy to implement.

Provide a theoretical performance guarantee?

Provide uniqueness of the minimizer in some sense?
A Generalization Bound

With high probability:

\[ R_{\text{true}} \leq R_{\text{emp},\theta} + \varepsilon \]

Want to minimize this, but can’t measure it

Can try to minimize this

Important factors:
- number of positive examples \( I \)
- number of negative examples \( K \)
- parameter \( \theta \)
- norm power \( p \)
- some notion of the complexity of the hypothesis space \( F:N(F,\varepsilon) \)
A Generalization Bound

The “true” objective function for which our algorithm is designed:

\[ R_{\text{true}}^p (f) := \left( \mathbb{E}_{x_- \sim D_-} \left( \mathbb{E}_{x_+ \sim D_+} 1_{[f(x_+)-f(x_-) \leq 0]} \right) \right)^{1/p} \]

\[ = \left\| P_{x_+ \sim D_+} (f(x_+)-f(x_-) \leq 0 \mid x_-) \right\|_{L_p(X_-, D_-)} \]

The empirical loss associated with \( R_{\text{true}}^p (f) \) is:

\[ R_{\text{emp}}^p (f) := \left( \frac{1}{K} \sum_{k=1}^{K} \left( \frac{1}{I} \sum_{i=1}^{I} 1_{[f(x_i^+)-f(x_k^-) \leq 0]} \right)^p \right)^{1/p} \]

A more general notion, incorporating a “margin”:

\[ R_{\text{emp}, \theta}^p (f) := \left( \frac{1}{K} \sum_{k=1}^{K} \left( \frac{1}{I} \sum_{i=1}^{I} 1_{[f(x_i^+)-f(x_k^-) \leq \theta]} \right)^p \right)^{1/p} \]
Theorem 2 (Generalization Bound): For all $\theta > 0, \varepsilon > 0$, $f \in F$

$$P_{S_+ \sim D^+, S_- \sim D^-} \left[ R_{\text{true}} \leq R_{\text{emp}, \theta} + \varepsilon \right]$$

$$\geq 1 - 2N \left( F, \frac{\varepsilon \theta}{8} \right) \exp \left[ -2 \left( \frac{\varepsilon}{4} \right)^{2p} K \right] + \exp \left[ -\frac{\varepsilon^2}{8} I \right]$$

Proof based on Rudin et al 05 and Rudin & Schapire 05, which was inspired by Koltchinskii and Panchenko 02, Cucker and Smale 00, Bousquet 02.

The bound says that generalization is possible!
Now we have a theoretical guarantee on performance!

Note: these kinds of bounds are usually proved with a symmetrization step, which does not work in our case. See (Agarwal et al. 04, Clemencon et al. 05, Usunier et al. 05)

The covering number approach fixes this. (Cucker and Smale 02, Rudin et al 05)
Outline of Talk

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

1) Deriving an Objective Function

2) The “P-Norm Push” Algorithm

3) A Generalization Bound

4) Uniqueness

A convex objective exists which concentrates near the top

Performs well on UCI data. Easy to implement.

Provide a theoretical performance guarantee?

Provide uniqueness of the minimizer in some sense?
Uniqueness \(\cong\)

- Want to show that the minimizer of \(F_p(\lambda)\) is somehow unique.

\[
\min_{? \in \mathbb{R}^n} F_p(\lambda) := \sum_{k=1}^{K} \left( \sum_{i=1}^{l} \exp(-f(x_i^+) + f(x_k^-)) \right)^p
\]

where \(f(x) = \sum_{j=1}^{n} \lambda_j h_j(x)\)

- Why is this tricky?
  - It’s not unique! At least with respect to \(\lambda\) (with no assumptions allowed on the \(h_j\)’s.)

- How can I keep things finite and get uniqueness?
  - use \(\{\exp(-f(x_i^+) + f(x_k^-))\}_{ik} \in \mathbb{R}^{IK}\)
Theorem 3 (Uniqueness)

Define

\[ Q' = \left\{ q'_ik = \exp(-f(x_i^+) + f(x_k^-)) \text{ for some } \lambda, \right\} \]

where \( f(x) = \sum_k \lambda_j h_j(x) \)

Then,

\[ q^* = \arg\min_{q' \in \text{closure}(Q')} \sum_k \left( \sum_i q'_{ik} \right)^p \]

and this determines \( q^* \) uniquely.

Proof based on convex duality for a class of Bregman distances (Della Pietra, Della Pietra, Lafferty 2002), (Collins, Schapire, Singer 2002)
Proof relies on finding a “magic function” to define the Bregman distance.

\[ \varphi(q) = -\sum_{ik} q_{ik} g(q_{ik}, q), \quad \text{where} \quad g(q_{ik}, q) := \ln \frac{q_{ik}}{p^{1/p} \left( \sum_{i'} q_{i'k} \right)^{(p-1)/p}} \]

(If p=1, this is the relative entropy between q and 1. Phew!)
Outline of Talk

- Introduction to the bipartite supervised ranking problem (Done)

Our Results:

1) Deriving an Objective Function

2) The “P-Norm Push” Algorithm

3) A Generalization Bound

4) Uniqueness

5) AdaBoost is a ranking algorithm

A convex objective exists which concentrates near the top

Performs well on UCI data. Easy to implement.

Provide a theoretical performance guarantee?

Provide uniqueness of the minimizer in some sense?
But Cynthia, AdaBoost is a *classification* algorithm…
Does it really do *ranking*?

Yes! And it’s as good as RankBoost!
(non-separable case)
AdaBoost is good for ranking (in addition to RankBoost). (It tends to achieve a high AUC value.)
Consider this movie ranking rule:

\[ h_c = \text{Movies that… are movies!} \quad h_c = 1 \text{ for every movie.} \]

Boogeyman, Spiderman, The Matrix, Titanic, Amelie

… everything satisfies this rule.

If we add it into the set of weak classifiers
(a very innocent assumption) and then run AdaBoost…

AdaBoost and RankBoost produce equally good solutions!
AdaBoost and RankBoost

- Define F-skew: it measures the imbalance of the loss between positive and negative examples.

\[ F\text{-skew}(\lambda) := \sum_{i \in \text{positives}} e^{-\langle M \lambda \rangle_i} - \sum_{k \in \text{negatives}} e^{-\langle M \lambda \rangle_k} \]

**AdaBoost and RankBoost Theorem** (R, Cortes, Mohri, Schapire, 05)

*Whenever the F-skew vanishes, AdaBoost converges to the minimum of RankBoost’s objective function. (Non-separable case)*

**Corollary:** *The F-skew vanishes whenever the constant hypothesis is included in the set of weak classifiers. So…*
Theorem (R, Cortes, Mohri, Schapire, 05)

Whenever the F-skew vanishes, AdaBoost and RankBoost converge to equally good AUC values. (nonseparable case.)
This is truly bizarre, but very good.
Thank you

Thanks to Rob Schapire, Eero Simoncelli, and Sinan Güntürk

Related Prior Work:

Dynamics of Boosting
(Rudin, Daubechies, Schapire: NIPS 04, JMLR 04)

Boosting Based on A Smooth Margin
(Rudin, Schapire, Daubechies: CoLT 04)

Margin-Based Ranking and Boosting Meet in the Middle
(Rudin, Cortes, Mohri, Schapire CoLT 05, Rudin and Schapire 05)

www.cns.nyu.edu/~rudin