Combined Regression and Ranking

D. Sculley
Google Pittsburgh
The Claim

- Many applications require models that give both:
  - good **regression** performance and
  - good **ranking** performance
Example: Predicting Star Ratings
Example: Click Prediction

Google search for "football tickets" showing sponsored links with options like "Steelers Football Tickets" and "NFL & College Football Tickets" from TicketLiquidator.com and TicketZoom.com.
Why not just use existing methods?
Standard Methods Can Fail Badly

- Rank-based models may do arbitrarily badly at regression
- Perfect regression implies perfect ranking, but...
- Even "good" regression can have bad ranking performance
Our Approach

- Novelty: optimize ranking and regression simultaneously
  
  - primary goal: try and get "best of both" performance
    - do as well at ranking as a ranking-only method
    - do as well at regression as a regression-only method
  
  - secondary goal: improved regression through ranking?

- We'll build this up in pieces
Supervised Regression (birds eye view)

Goal: learn a model $w$ that predicts a real valued target $y$

Examples:
- Least mean squares
- Ridge Regression
- LASSO

Often solved using empirical risk minimization
Supervised Regression (review)

$$\min_{w \in \mathbb{R}^m} L(w, D) + \frac{\lambda}{2} ||w||_2^2$$
Supervised Regression (review)

\[
\min_{w \in \mathbb{R}^m} L(w, D) + \frac{\lambda}{2} \|w\|_2^2
\]

- **Loss Function**
- **Examples**: Squared Loss, Logistic Loss, etc.
- **Regularization Term**
Supervised Ranking (review)

- Goal: learn a model $w$ that puts unseen data in the correct preference order

- Several known methods:
  - RankSVM (Joachims, 2002)
  - Voted Perceptron variant (Elsas et al., 2008)
  - Boosting variants: AdaRank-MAP, AdaRank-NDCG (Xu and Li, 2007)
  - Listwise approach (Cao et al., 2007)
Supervised Ranking (review)

$$\min_{w \in \mathbb{R}^m} L(w, P) + \frac{\lambda}{2} \|w\|_2^2$$
Supervised Ranking (review)

\[
\min_{w \in \mathbb{R}^m} L(w, P) + \frac{\lambda}{2} \|w\|_2^2
\]

Candidate Pairs: pairs \((a, b)\) of comparable examples with different ranks
Supervised Ranking (review)

Google search for "supervised ranking"

- [PDF] Supervised Rank Aggregation
  - File Format: PDF/Adobe Acrobat - Quick View
  - by YT Liu - Cited by 27 - Related articles
  - Supervised Rank Aggregation, in which learning is formalized an ... meta-searches show that
  - Supervised Rank Aggregation can ...

- Supervised rank aggregation
  - by YT Liu - 2007 - Cited by 27 - Related articles
  - We refer to the approach as Supervised Rank Aggregation. We set up a general framework
  - for conducting Supervised Rank Aggregation, in which learning is ...
  - portal.acm.org/citation.cfm?id=1242638 - Similar

- Supervised ranking in open-domain text summarization
  - by T Nomoto - 2002 - Cited by 3 - Related articles
  - Supervised ranking in open-domain text summarization. Full text, Publisher Site, Pdf (142 KB).
  - Source, Annual Meeting of the ACL archive ...
  - portal.acm.org/citation.cfm?id=1073161

- [PDF] Supervised Ranking in Open-Domain Text Summarization
  - by T Nomoto - Cited by 3 - Related articles
  - 2 Supervised Ranking with Probabilistic. Decision Tree. One technical problem associated
  - with the use of a decision tree as a summarizer is that it is not ...
  - www.ldc.upenn.edu/acl/P/P02/P02-1059.pdf

Candidate Pairs: pairs (a,b) of comparable examples with different ranks
Supervised Ranking (review)

\[
\min_{\mathbf{w} \in \mathbb{R}^m} L(\mathbf{w}, P) + \frac{\lambda}{2} \| \mathbf{w} \|_2^2
\]

Warning: P is quadratic in |D|
• Joint optimization...
Combined Ranking and Regression

$$\min_{w \in \mathbb{R}^m} \alpha L(w, D) + (1 - \alpha) L(w, P) + \frac{\lambda}{2} \|w\|_2^2$$
Combined Ranking and Regression

\[
\min_{w \in \mathbb{R}^m} \alpha L(w, D) + (1 - \alpha) L(w, P) + \frac{\lambda}{2} \|w\|^2
\]
Combined Ranking and Regression

$$\min_{\mathbf{w} \in \mathbb{R}^m} \alpha L(\mathbf{w}, D) + (1 - \alpha) L(\mathbf{w}, P) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

Convexity Maintained
What about dealing with size of $P$? This is quadratic in $|D|$. 
Efficient Sampling from $P$

- We don't want to look at $O(n^2)$ training pairs
- How to sample pairs from $P$?

- Fastest solution is to index the training data:
  - $O(\log|Q| + \log|Y|)$ in general
  - $O(1)$ for common scenarios

- When data is too large to index, can use rejection sampling
Solving CRR Efficiently

Repeat...

Flip biased coin

Randomly pick one example $x$ from $D$

Randomly pick one pair $a, b$ from $P$

$x = a - b$

Update model based on $\text{Loss}(w, x)$
Scalability

- Like other stochastic gradient descent algorithms, CRR is fast for large data.

- RCV1 experiments
  - 780,000 training examples
  - Less than 3 CPU sec's on normal laptop
Non-linear Models

- CRR optimization problem is defined using a linear model with
- If we want non-linearity, use a trick from Balcan and Blum:
  - Pick a set of $k$ reference examples $r_1, \ldots, r_k$
  - Map each example $x$ into a new feature space of dimension $k$
    - Value for feature $i$ in new space is $\text{kernel}(x, r_i)$
- Still efficient
Experimental Overview

- Data sets:
  - RCV1 text classification
  - LETOR learning to rank benchmark data
  - Click prediction data for sponsored search (private)

- Comparison methods:
  - Regression-only, Ranking-only
  - Parameters tuned with cross validation on training data or on separate validation data

- Evaluation metrics:
  - Mean Squared Error (MSE)
  - AUC Loss (1 - Area Under ROC Curve)
  - Normalized Discounted Cumulative Gain (NDCG)
  - Mean Average Precision (MAP)
RCV1 Setup

- Benchmark text mining data set
- Tested 40 per-topic tasks
- ~780k training examples
- ~23k test examples
- ~50k sparse features
- Some topics contain extreme minority class distributions, with only 0.02% "positive"
- Used logistic loss on \{0, 1\} targets
RCV1 Ranking Results
RCV1 Regression Results

![Graph showing regression results with lines for Ranking Only, Regression Only, and CRR.]
RCV1 Results

- CRR achieves "best of both" metrics on 16 out of 40 tasks
  - Within 0.001 of best on 19 additional tasks
  - Always gives best performance on at least one of the two metrics

- Adding rank-based constraints can help regression:
  - CRR out-performs regression-only on MSE on 20 of 33 extreme minority class topics
  - Gives equal performance on remainder
Why Would Ranking Help Regression?

- Rank-based constraints are informative, especially when observations are rare.

- Imagine you had two biased coins:
  - A comes up heads with probability 0.02
  - B comes up heads with probability 0.03

- Knowing that coin C is between A and B is extremely helpful if we don't have much other data.
LETOR Experiments

- LETOR: benchmark learning to rank data
- Tasks with multiple relevance levels: 1, 2, or 3 stars
- Used squared loss; regression predicts ordinal values
LETOR Ranking Results

LETOR Results

Higher is Better

MAP

NDGC

Regression-only
Rank-only
CRR
LETOR Regression Results

![LETOR Results Graph]

- **Regression-only**
- **Rank-only**
- **CRR**

MSE (Lower is Better) vs. MSE
Click Prediction Experiments

- Test data set of several million ads
- Labels of "clicked" and "not clicked"
- Very high dimensional feature space
- Logistic loss used
## Click Prediction Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Sq. Error</th>
<th>AUC Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking-only</td>
<td>0.0935</td>
<td>0.1325</td>
</tr>
<tr>
<td>Regression-only</td>
<td>0.0840</td>
<td>0.1334</td>
</tr>
<tr>
<td>CRR</td>
<td>0.0840</td>
<td>0.1325</td>
</tr>
</tbody>
</table>
Click Prediction Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Sq. Error</th>
<th>AUC Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking-only</td>
<td>0.0935</td>
<td>0.1325</td>
</tr>
<tr>
<td>Regression-only</td>
<td>0.0840</td>
<td>0.1334</td>
</tr>
<tr>
<td>CRR</td>
<td>0.0840</td>
<td>0.1325</td>
</tr>
</tbody>
</table>

11% better than ranking-only
0.8% better than regression-only

Improvements are statistically significant
How sensitive is the tradeoff parameter alpha?
Combined Ranking and Regression

\[
\min_{\mathbf{w} \in \mathbb{R}^m} \alpha L(\mathbf{w}, D) + (1 - \alpha) L(\mathbf{w}, P) + \frac{\lambda}{2} \|\mathbf{w}\|^2_2
\]
Looking at Tradeoff Parameter, \( \alpha \)

Good results across range of intermediate values
Wrapping Up...

- Combined Ranking and Regression often gives "best of both" performance

- This algorithm uses pairwise method for rank-based component

- Simple, scalable, and robust

- Promising area for additional work
  - consider joint optimizations including MAP or NDCG optimization for ranking component
Thank you!

Questions?

Open Source Code: http://code.google.com/p/sofia-ml

Email: dsculley@google.com
<table>
<thead>
<tr>
<th>Task</th>
<th>% Positive</th>
<th>Regression AUC Loss</th>
<th>Regression MSE</th>
<th>Ranking AUC Loss</th>
<th>Ranking MSE</th>
<th>CRR AUC Loss</th>
<th>CRR MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>E141</td>
<td>0.05%</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.203</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>GOBIT</td>
<td>0.06%</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.162</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>E61</td>
<td>0.06%</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.320</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>GTOUR</td>
<td>0.10%</td>
<td>0.030</td>
<td>0.001</td>
<td>0.005</td>
<td>0.245</td>
<td>0.005</td>
<td>0.001</td>
</tr>
<tr>
<td>C331</td>
<td>0.13%</td>
<td>0.003</td>
<td>0.001</td>
<td>0.001</td>
<td>0.205</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>E143</td>
<td>0.15%</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.296</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>G152</td>
<td>0.15%</td>
<td>0.005</td>
<td>0.001</td>
<td>0.003</td>
<td>0.239</td>
<td>0.003</td>
<td>0.001</td>
</tr>
<tr>
<td>G155</td>
<td>0.16%</td>
<td>0.007</td>
<td>0.002</td>
<td>0.004</td>
<td>0.223</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td>E411</td>
<td>0.17%</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.289</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>C313</td>
<td>0.18%</td>
<td>0.047</td>
<td>0.002</td>
<td>0.014</td>
<td>0.281</td>
<td>0.016</td>
<td>0.002</td>
</tr>
<tr>
<td>E311</td>
<td>0.19%</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
<td>0.311</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>C32</td>
<td>0.19%</td>
<td>0.019</td>
<td>0.002</td>
<td>0.012</td>
<td>0.180</td>
<td>0.013</td>
<td>0.002</td>
</tr>
<tr>
<td>G157</td>
<td>0.19%</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
<td>0.254</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>C16</td>
<td>0.21%</td>
<td>0.022</td>
<td>0.002</td>
<td>0.012</td>
<td>0.234</td>
<td>0.013</td>
<td>0.002</td>
</tr>
<tr>
<td>GWELF</td>
<td>0.22%</td>
<td>0.010</td>
<td>0.002</td>
<td>0.005</td>
<td>0.236</td>
<td>0.006</td>
<td>0.002</td>
</tr>
<tr>
<td>E513</td>
<td>0.23%</td>
<td>0.004</td>
<td>0.002</td>
<td>0.003</td>
<td>0.300</td>
<td>0.003</td>
<td>0.001</td>
</tr>
<tr>
<td>E14</td>
<td>0.28%</td>
<td>0.008</td>
<td>0.003</td>
<td>0.003</td>
<td>0.281</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>C173</td>
<td>0.33%</td>
<td>0.005</td>
<td>0.003</td>
<td>0.004</td>
<td>0.237</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>E121</td>
<td>0.41%</td>
<td>0.007</td>
<td>0.004</td>
<td>0.004</td>
<td>0.261</td>
<td>0.005</td>
<td>0.003</td>
</tr>
<tr>
<td>GENT</td>
<td>0.46%</td>
<td>0.014</td>
<td>0.004</td>
<td>0.008</td>
<td>0.126</td>
<td>0.008</td>
<td>0.004</td>
</tr>
<tr>
<td>C34</td>
<td>0.52%</td>
<td>0.018</td>
<td>0.005</td>
<td>0.011</td>
<td>0.231</td>
<td>0.012</td>
<td>0.004</td>
</tr>
<tr>
<td>GHEA</td>
<td>0.85%</td>
<td>0.007</td>
<td>0.008</td>
<td>0.005</td>
<td>0.140</td>
<td>0.006</td>
<td>0.006</td>
</tr>
<tr>
<td>C183</td>
<td>0.87%</td>
<td>0.013</td>
<td>0.008</td>
<td>0.009</td>
<td>0.275</td>
<td>0.010</td>
<td>0.006</td>
</tr>
<tr>
<td>GDEF</td>
<td>1.01%</td>
<td>0.015</td>
<td>0.009</td>
<td>0.009</td>
<td>0.208</td>
<td>0.009</td>
<td>0.007</td>
</tr>
<tr>
<td>C42</td>
<td>1.48%</td>
<td>0.009</td>
<td>0.010</td>
<td>0.006</td>
<td>0.242</td>
<td>0.007</td>
<td>0.008</td>
</tr>
<tr>
<td>E211</td>
<td>1.76%</td>
<td>0.013</td>
<td>0.011</td>
<td>0.010</td>
<td>0.245</td>
<td>0.010</td>
<td>0.009</td>
</tr>
<tr>
<td>E51</td>
<td>2.77%</td>
<td>0.025</td>
<td>0.019</td>
<td>0.019</td>
<td>0.280</td>
<td>0.021</td>
<td>0.016</td>
</tr>
<tr>
<td>M12</td>
<td>3.16%</td>
<td>0.010</td>
<td>0.015</td>
<td>0.008</td>
<td>0.288</td>
<td>0.009</td>
<td>0.014</td>
</tr>
<tr>
<td>C24</td>
<td>3.98%</td>
<td>0.031</td>
<td>0.027</td>
<td>0.025</td>
<td>0.157</td>
<td>0.026</td>
<td>0.024</td>
</tr>
<tr>
<td>GDIP</td>
<td>4.34%</td>
<td>0.019</td>
<td>0.023</td>
<td>0.017</td>
<td>0.188</td>
<td>0.018</td>
<td>0.022</td>
</tr>
<tr>
<td>M13</td>
<td>6.89%</td>
<td>0.007</td>
<td>0.018</td>
<td>0.007</td>
<td>0.221</td>
<td>0.007</td>
<td>0.018</td>
</tr>
<tr>
<td>GPOL</td>
<td>7.11%</td>
<td>0.021</td>
<td>0.031</td>
<td>0.020</td>
<td>0.175</td>
<td>0.021</td>
<td>0.031</td>
</tr>
<tr>
<td>C152</td>
<td>8.34%</td>
<td>0.026</td>
<td>0.036</td>
<td>0.023</td>
<td>0.178</td>
<td>0.024</td>
<td>0.035</td>
</tr>
<tr>
<td>C151</td>
<td>10.22%</td>
<td>0.010</td>
<td>0.024</td>
<td>0.009</td>
<td>0.188</td>
<td>0.009</td>
<td>0.025</td>
</tr>
<tr>
<td>M14</td>
<td>10.98%</td>
<td>0.005</td>
<td>0.021</td>
<td>0.004</td>
<td>0.115</td>
<td>0.004</td>
<td>0.022</td>
</tr>
<tr>
<td>ECAT</td>
<td>14.90%</td>
<td>0.033</td>
<td>0.054</td>
<td>0.030</td>
<td>0.188</td>
<td>0.031</td>
<td>0.053</td>
</tr>
<tr>
<td>C15</td>
<td>18.05%</td>
<td>0.013</td>
<td>0.038</td>
<td>0.013</td>
<td>0.132</td>
<td>0.013</td>
<td>0.037</td>
</tr>
<tr>
<td>MCAT</td>
<td>25.41%</td>
<td>0.011</td>
<td>0.039</td>
<td>0.010</td>
<td>0.113</td>
<td>0.010</td>
<td>0.043</td>
</tr>
<tr>
<td>GCAT</td>
<td>30.11%</td>
<td>0.012</td>
<td>0.043</td>
<td>0.012</td>
<td>0.062</td>
<td>0.012</td>
<td>0.046</td>
</tr>
<tr>
<td>CCAT</td>
<td>46.59%</td>
<td>0.022</td>
<td>0.067</td>
<td>0.022</td>
<td>0.073</td>
<td>0.022</td>
<td>0.070</td>
</tr>
</tbody>
</table>
Click Prediction Results

<table>
<thead>
<tr>
<th>Method</th>
<th>AdSet1 AUC Loss</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>0.133</td>
<td>0.084</td>
</tr>
<tr>
<td>Ranking</td>
<td>0.132</td>
<td>0.094</td>
</tr>
<tr>
<td>CRR</td>
<td>0.132</td>
<td>0.084</td>
</tr>
</tbody>
</table>

0.8% improvement in AUC loss with same MSE
Difference is statistically significant