Optimizing Estimated Loss Reduction for Active Sampling in Rank Learning : DiffLoss

Presented by Pinar Donmez
joint work with Jaime G. Carbonell

Language Technologies Institute
Carnegie Mellon University

ICML ‘08, Helsinki, Finland, June 7 2008

Road Map

- The Challenge: Active Rank Learning
- Related Work
- DiffLoss: New Method for Active Learning for RankSVM and RankBoost
- Results: DiffLoss vs. Margin-Based and Random Sampling
- Conclusion

Active Rank Learning: Why do we care?

- Challenge: Labeling for rank learning
  - requires eliciting relative ordering over a set of alternatives
    - costly
    - time-consuming
    - extensive human effort
- Numerous applications
  - document retrieval
  - collaborative filtering
  - product rating...


Active Rank Learning: How do we approach?

- Motivation
  - an optimal active learner samples those with the lowest estimated expected error on the test set (Roy & McCallum, 2001)
  - impractical for large-scale ranking problems even with efficient re-training
- Our solution:
  - estimate this expectation ***without any re-training***
  - based on the likelihood of the change of the current hypothesis
  - greater chance to learn the true hypothesis faster

Related Work

- Margin-based Sampling (Brinker, 2004; Yu, 2005)
  - margin := minimum difference of scores between two instances in the ranked order
  - selects the examples with minimum margin
    - pro: general, and simple to implement
    - con: similar instances with the same rank label may have minimum margin
- Divergence-based Sampling (Amini et al, 2006)
  - similar to query-by-committee sampling
  - selects instances at which two ranking functions maximally disagree
    - pro: theoretical justification
    - con: effective only when provided with a sufficiently large initial labeled set

DiffLoss for RankSVM

- Assume $\vec{x} \in U$ is added to training set with $y \in Y$
- Total loss on pairs that include $\vec{x}$ is:

$$D(\vec{x}, \vec{w}) = \sum_{j=1}^{n} [1 - z_j \langle \vec{w}, \vec{x}_j - \vec{x} \rangle]_+$$

Label for the difference vector $\vec{x}_j - \vec{x}$
+1 if $\vec{x}_j \succ \vec{x}$ and -1 otherwise

Linear RankSVM solution:

$$f(\vec{x}) = \langle \vec{w}, \vec{x} \rangle$$

- $\vec{x}_j$ `s: training instances with a different label than y
- n is the # of such instances

- Objective function to be minimized then becomes:

$$\min_{\vec{w}} \left\{ \lambda \|\vec{w}\|^2 + \sum_{k=1}^{K} \left[1 - z_k \left\langle \vec{w}, \vec{x}_k^1 - \vec{x}_k^2 \right\rangle \right]_+ + D(\vec{x}, \vec{w}) \right\}$$

- Assume the current ranking function is $f(\vec{x}) = \langle \vec{w}^*, \vec{x} \rangle$
- There are two possible cases:

$$\vec{w}^* = \operatorname{argmin}_{\vec{w}} D(\vec{w}, \vec{x}) \qquad \text{or} \qquad \vec{w}^* \neq \operatorname{argmin}_{\vec{w}} D(\vec{w}, \vec{x})$$

- Assume $\hat{\vec{w}} = \operatorname{argmin}_{\vec{w}} D(\vec{w}, \vec{x}) = \min_{\vec{w}} \sum_{j=1}^{n} \left[1 - z_j \left\langle \vec{w}, \vec{x}_j - \vec{x} \right\rangle \right]_+$

- Derivative w.r.t $\vec{w}$ at a single pair

$$\Delta \vec{w}_j = \begin{cases} 0 & \text{if } z_j \left\langle \vec{w}, \vec{x}_j - \vec{x} \right\rangle \geq 1 \\ -z_j (\vec{x}_j - \vec{x}) & \text{if } z_j \left\langle \vec{w}, \vec{x}_j - \vec{x} \right\rangle < 1 \end{cases}$$

DiffLoss for RankSVM Final Selection

- Substitute $\vec{w}^*$ in

  to estimate $\|\vec{w}^* - \hat{\vec{w}}\| = \|\Delta\vec{w}\|$
- Magnitude of the total derivative

$$\|\Delta\vec{w}\|_y = \sum_j \|\Delta\vec{w}_j\| = \sum_{j=1}^{n} \begin{cases} 0 & \text{if } z_j \langle \vec{w}^*, \vec{x}_j - \vec{x} \rangle \geq 1 \\ \|-z_j(\vec{x}_j - \vec{x})\| & \text{if } z_j \langle \vec{w}^*, \vec{x}_j - \vec{x} \rangle < 1 \end{cases}$$

- $\|\Delta\vec{w}\|_y$: ability of $\vec{x}$ to change the current ranker if added into training
- Sample $\vec{x}^* = \operatorname*{argmax}_{\vec{x} \in U} \sum_{y \in Y} \hat{P}(y \mid \vec{x}) \|\Delta\vec{w}\|_y$

DiffLoss for RankBoost

- Estimate how the current ranker would change if $\vec{x} \in U$ was in the training set

- Estimate by the ranking loss on the new pairs that include $\vec{x} \in U$

- Ranking loss w.r.t D_{T+1} is (Freund et al., 2003):

$$D_{T+1}(\vec{x}^1, \vec{x}^2) = D_1(\vec{x}^1, \vec{x}^2) \frac{exp(H(\vec{x}^2) - H(\vec{x}^1))}{\prod_t Z_t}$$

DiffLoss for RankBoost Final Selection

- Ranking loss on the new pairs:

$$\Delta L(\vec{x}, y=1) = \sum_{\vec{x}^j, \vec{x}} \frac{exp(H(\vec{x}^j) - H(\vec{x}))}{\prod_t Z_t} I(H(\vec{x}^j) \geq H(\vec{x}))$$

- $\Delta L(\vec{x}, y=1)$ estimates the change in the current ranker if $\vec{x} \in U$ was sampled
- Sample the instance with the highest loss differential:

$$\vec{x}^* = \underset{\vec{x} \in U}{\operatorname{argmax}} \left\{ \hat{P}(y=1 \mid \vec{x}) \Delta L(\vec{x}, y=1) + \hat{P}(y=-1 \mid \vec{x}) \Delta L(\vec{x}, y=-1 \right\}$$

Data & Settings

Dataset	Query size	Docs/Q	%Rel
TD2003	50	983.42	1%
TD2004	75	988.93	0.6%

- TREC 2003 and TREC 2004 topic distillation datasets in LETOR
  - Binary relevance
- Start with 16 docs/query (1 relevant & 15 non-relevant)
- Select 5 docs/query at each iteration
- 25 iterations

Performance Measures

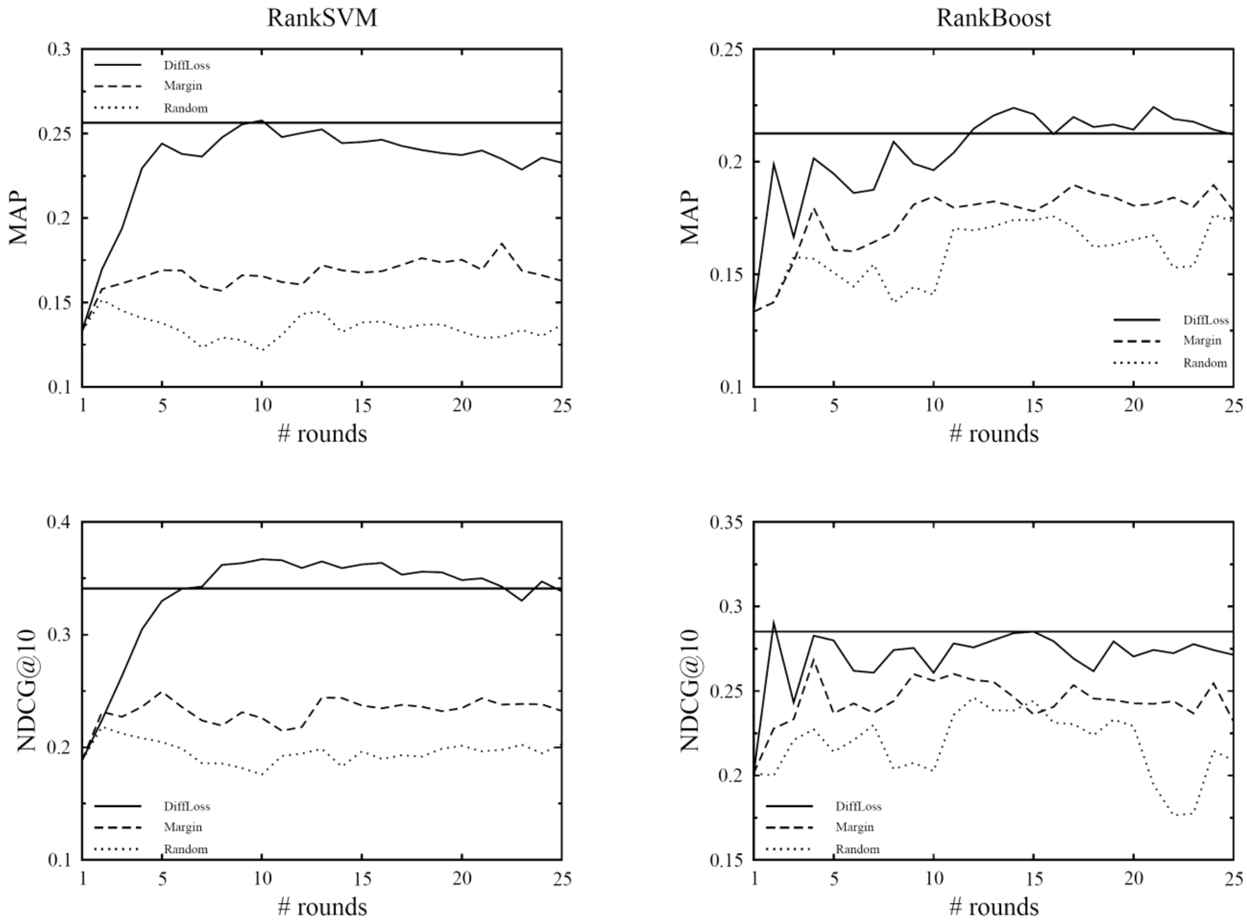
- MAP (Mean Average Precision)

$$AP = \frac{\sum_{n=1}^{N}(P(r) * rel(r))}{\#\ \text{total relevant documents for this query}}$$

  - MAP is the average of AP values for all queries
- NDCG (Normalized Discounted Cumulative Gain)
  - The impact of each relevant document is discounted as a function of rank position

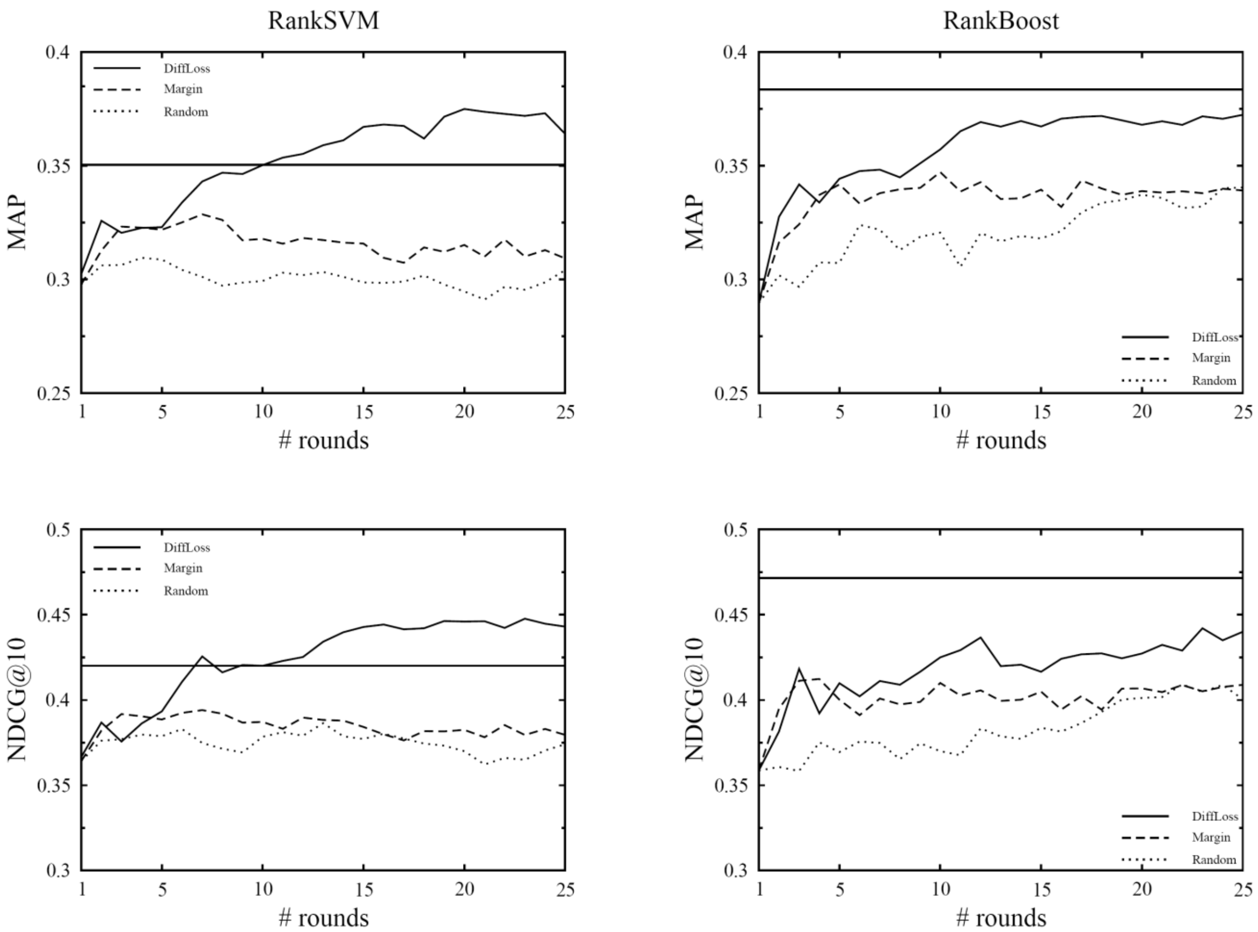
$$NDCG@n = Z_n \sum_{j=1}^{n} \frac{2^{r(j)}-1}{log(1+j)}$$

Results on TREC03



* Horizontal line indicates the performance if all the data is used as the training set.

Results on TREC04



Results at a Glance

- DiffLoss:
  - significantly superior over the entire operating range ($p<0.0001$).
  - achieves 30% relative improvement over the margin-based sampling on TREC03.
  - using RankSVM reaches the optimal performance after ~10 rounds.
  - using RankBoost reaches 90-95% of the optimal performance after ~10 rounds.

Conclusion

- A new active sampling framework for rank learning
- Sample instances with the largest expected loss differential
- significantly faster learning rate compared to baselines
- In the future, we plan to focus on
  - sampling by directly optimizing performance metrics
  - automatically determining when to stop sampling

THE END!