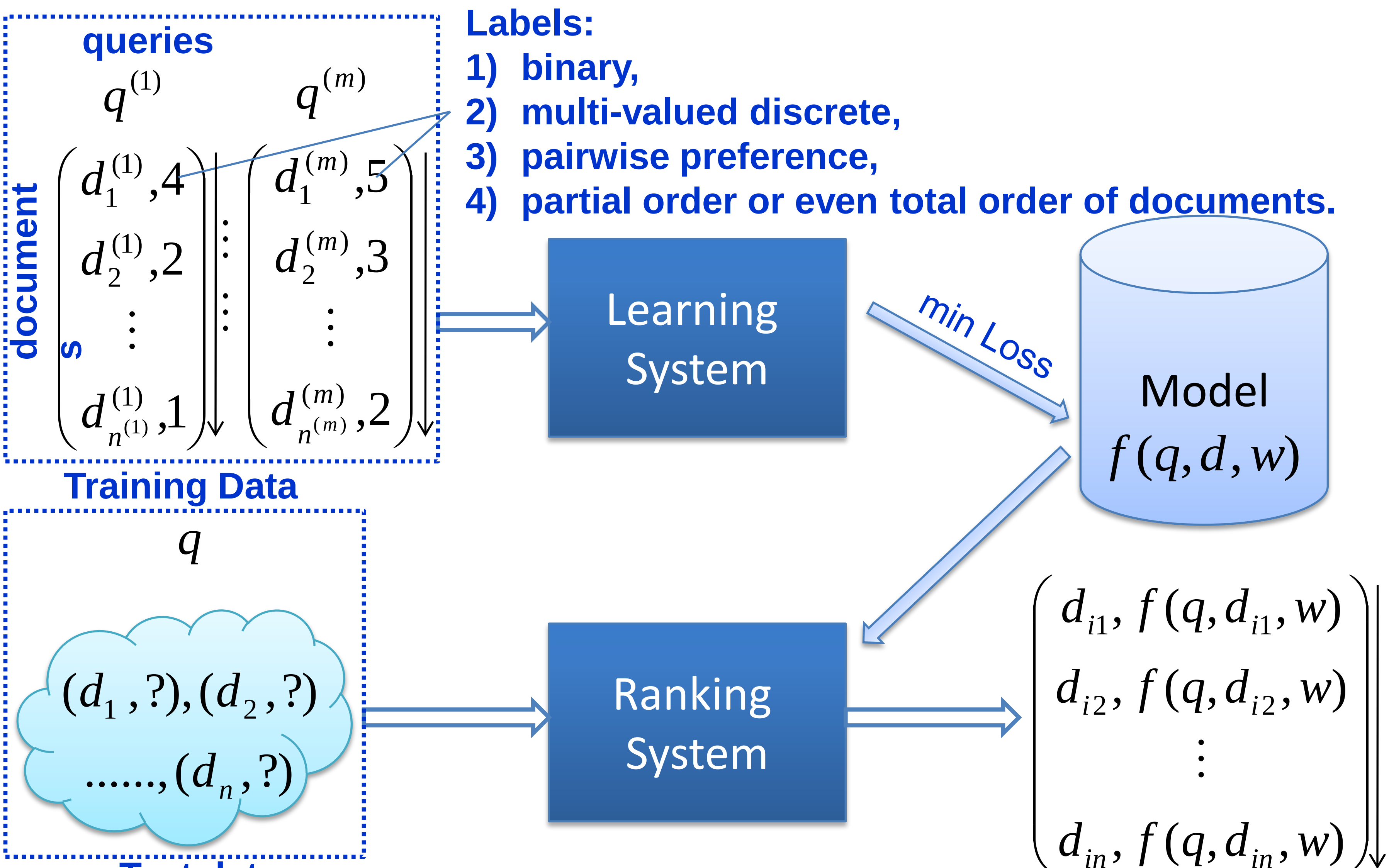
Microsoft Research

Listwise Approach to Learning to Rank – Theory and Algorithm

Fen Xia*, Tie-Yan Liu
Jue Wang, Wensheng Zhang and Hang Li

Microsoft Research Asia
Chinese Academy of Sciences

Learning to Rank for Information Retrieval



State-of-the-art Approaches

- Pointwise: (Ordinal) regression / classification
  - Pranking, MCRank, etc.
- Pairwise: Preference learning
  - Ranking SVM, RankBoost, RankNet, etc.
- Listwise: Taking the entire set of documents associated with a query as the learning instance.
  - Direct optimization of IR measure
    - AdaRank, SVM-MAP, SoftRank, LambdaRank, etc.
  - Listwise loss minimization
    - RankCosine, ListNet, etc.

Motivations

- The listwise approach captures the ranking problem in a conceptually more natural way and performs better than other approaches on many benchmark datasets.
- However, the listwise approach lacks of theoretical analysis.
  - Existing work focuses more on algorithm and experiments, than theoretical analysis.
  - While many existing theoretical results on regression and classification can be applied to the pointwise and pairwise approaches, the theoretical study on the listwise approach is not sufficient.

Our Work

- Take listwise loss minimization as an example, to perform theoretical analysis on the listwise approach.
  - Give a formal definition of the listwise approach.
  - Conduct theoretical analysis on listwise ranking algorithms in terms of their loss functions.
  - Propose a novel listwise ranking method with good loss function.
  - Validate the correctness of the theoretical findings through experiments.

Listwise Ranking

- Input space: *X*
  - Elements in *X* are sets of objects to be ranked
- Output space: *Y*
  - Elements in *Y* are permutations of objects
- Joint probability distribution: P_{XY}
- Hypothesis space: *H*
  - $\mathbf{h} \in H : X \to Y$
- Expected loss

$$R(\mathbf{h}) = \int_{X \times Y} l(\mathbf{h}(\mathbf{x}), \mathbf{y}) dP(\mathbf{x}, \mathbf{y})$$

Empirical loss

$$R_S(\mathbf{h}) = \frac{1}{m} \sum_{i=1}^{m} l(\mathbf{h}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}).$$

True Loss in Listwise Ranking

- To analyze the theoretical properties of listwise approach, the "true" loss of ranking is to be defined.
  - The true loss describes the difference between a given ranked list (permutation) and the ground truth ranked list (permutation).
- Ideally, the "true" loss should be cost-sensitive, but for simplicity, we start with the investigation of the "0-1" loss.
  - $l(\mathbf{h}(\mathbf{x}), \mathbf{y}) = \begin{cases} 1, & \text{if } \mathbf{h}(\mathbf{x}) \neq \mathbf{y} \\ 0, & \text{if } \mathbf{h}(\mathbf{x}) = \mathbf{y}, \end{cases}$

Surrogate Loss in Listwise Ranking

- Widely-used ranking function
  - $\mathbf{h}(\mathbf{x}^{(i)}) = sort(g(x_1^{(i)}), \ldots, g(x_{n_i}^{(i)})).$
- Corresponding empirical risk
  - $R_S(\mathbf{g}) = \frac{1}{m}\sum_{i=1}^{m} l(sort(g(x_1^{(i)}), \ldots, g(x_{n_i}^{(i)})), \mathbf{y}^{(i)})$
- Challenges
  - Due to the sorting function and the 0-1 loss, the empirical loss is non-differentiable w.r.t. g(x).
  - To tackle the problem, a surrogate loss is used.

$$R_S^{\phi}(\mathbf{g}) = \frac{1}{m}\sum_{i=1}^{m} \phi(\mathbf{g}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})$$

Surrogate Listwise Loss Minimization

- RankCosine and ListNet can be well fitted into the framework of surrogate loss minimization.
  - Cosine Loss (RankCosine, IPM 2007)

$$\phi(\mathbf{g}(\mathbf{x}),\mathbf{y}) = \frac{1}{2}\left(1 - \frac{\psi_{\mathbf{y}}(\mathbf{x})^T \mathbf{g}(\mathbf{x})}{\|\psi_{\mathbf{y}}(\mathbf{x})\|\|\mathbf{g}(\mathbf{x})\|}\right).$$

  - Cross Entropy Loss (ListNet, ICML 2007)

$$\phi(\mathbf{g}(\mathbf{x}),\mathbf{y}) = D(P(\pi|\mathbf{x};\psi_{\mathbf{y}})||P(\pi|\mathbf{x};\mathbf{g}))$$

- A new loss function
  - Likelihood loss (ListMLE, proposed in this paper)

$$\phi(\mathbf{g}(\mathbf{x}),\mathbf{y}) = -\log P(\mathbf{y}|\mathbf{x};\mathbf{g}) \qquad P(\mathbf{y}|\mathbf{x};\mathbf{g}) = \prod_{i=1}^{n} \frac{\exp(g(x_{\mathbf{y}(i)}))}{\sum_{k=i}^{n} \exp(g(x_{\mathbf{y}(k)}))}$$

Analysis on Surrogate Loss

- Continuity, differentiability and convexity
- Computational efficiency
- Statistical consistency
- Soundness

These properties have been well studied in classification, but not sufficiently in ranking.

Continuity, Differentiability, Convexity, Efficiency

Loss	Continuity	Differentiability	Convexity	Efficiency
Cosine Loss (RankCosine)	√	√	X	O(n)
Cross-entropy loss (ListNet)	√	√	√	O(n·n!)
Likelihood loss (ListMLE)	√	√	√	O(n)

Statistical Consistency

- When minimizing the expected surrogate loss $R^{\phi}(g)$ is equivalent to minimizing the expected 0-1 loss R(h) (which solution is Bayes ranker y^*), we say the surrogate loss function is consistent.

Theorem. *Let $\phi_{\mathbf{y}}(\mathbf{g})$ be an order sensitive loss function on $\Omega \subset R^n$. $\forall n$ objects, if its permutation probability space is order preserving with respect to $n-1$ objective pairs $(j_1, j_2), (j_2, j_3), \cdots, (j_n - 1, j_n)$. Then the loss $\phi_{\mathbf{y}}(\mathbf{g})$ is consistent.*

The perfect ranking of an object is inherently determined by its own.

Minimum $\phi_y(g)$ is achieved when sorting g(x) results in the same permutation with a given y.

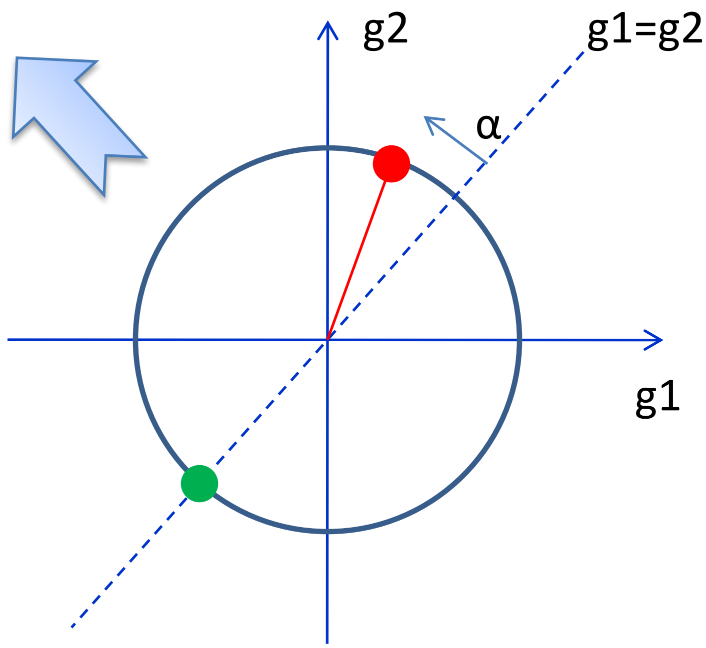
Statistical Consistency (3)

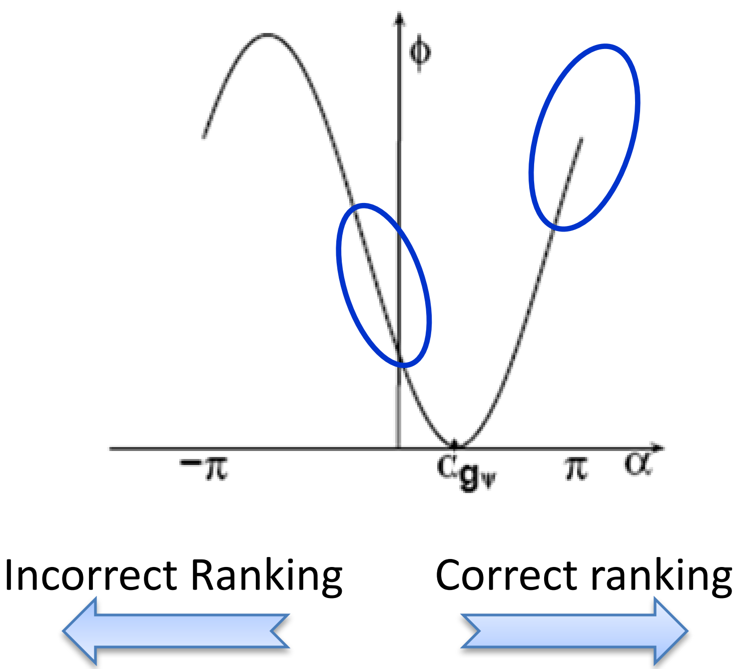
- It can be proven
  - Cosine Loss is statistically consistent.
  - Cross entropy loss is statistically consistent.
  - Likelihood loss is statistically consistent.

  - For detailed proof, please refer to the paper.

Soundness

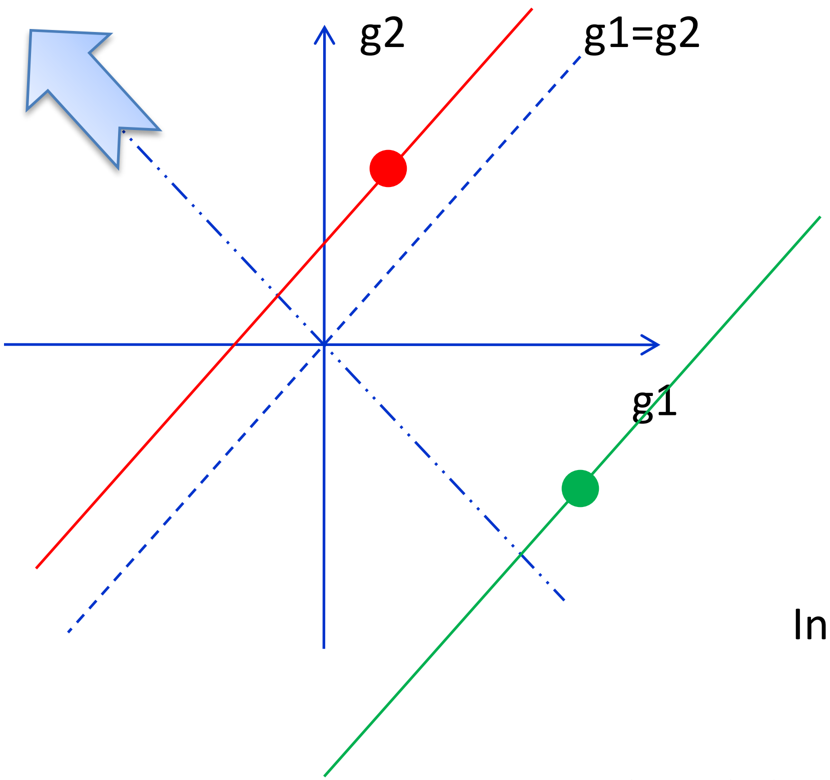
- Cosine loss is not very sound
  - Suppose we have two documents D2 ⊳ D1.

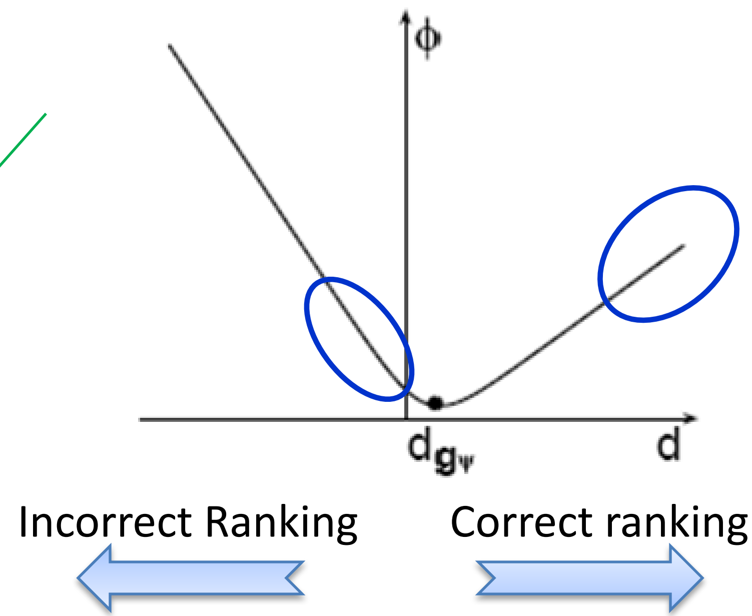




Soundness (2)

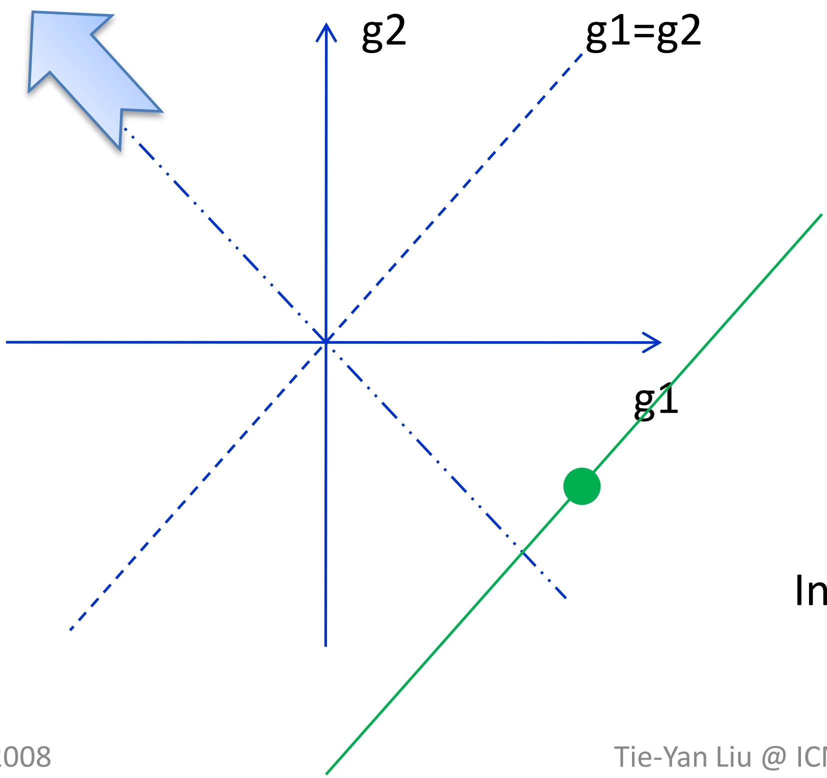
- Cross entropy loss is not very sound
  - Suppose we have two documents D2 ⊳ D1.

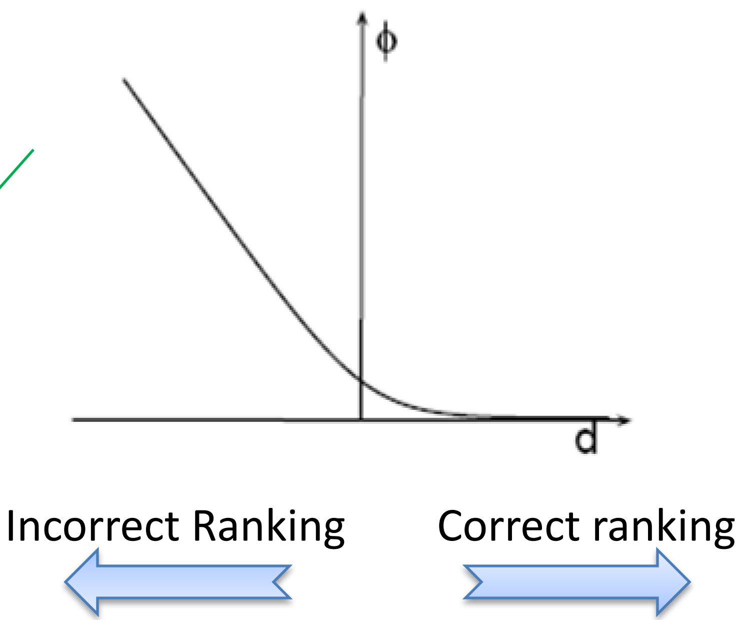




Soundness (3)

- Likelihood loss is sounder
  - Suppose we have two documents D2 ▷ D1.





Discussions

- All three losses can be minimized using common optimization technologies. (continuity and differentiability)
- When the number of training samples is very large, the model learning can be effective. (consistency)
- The cross entropy loss and the cosine loss are both sensitive to the mapping function. (soundness)
- The cost of minimizing the cross entropy loss is high. (complexity)
- The cosine loss is sensitive to the initial setting of its minimization. (convexity)
- **The likelihood loss is the best among the three losses.**

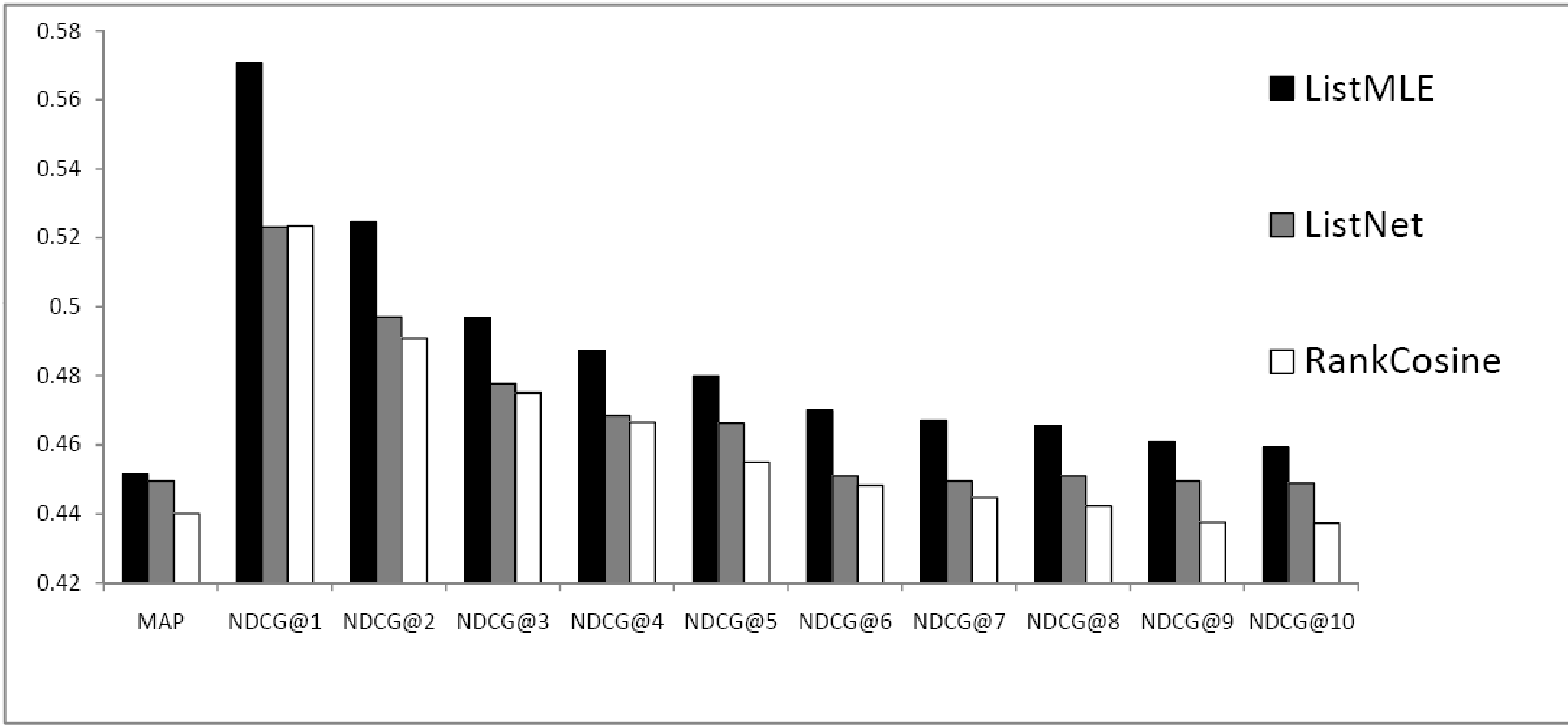
Experimental Verification

- Synthetic data
  - Different mapping function (log, sqrt, linear, quadratic, and exp)
  - Different initial setting of the gradient descent algorithm (report the mean and var of 50 runs)
- Real data
  - OHSUMED dataset in the LETOR benchmark

Experimental Results on Synthetic Data

Algorithm	Accuracy	MAP
ListMLE	0.92 ± 0.011	0.999 ± 0.002
ListNet-log	0.905 ± 0.010	0.999 ± 0.002
ListNet-sqrt	0.917 ± 0.009	0.999 ± 0.002
ListNet-l	0.767 ± 0.021	0.995 ± 0.003
ListNet-q	0.868 ± 0.028	0.999 ± 0.002
ListNet-exp	0.832 ± 0.074	0.997 ± 0.004
RankCosine-log	0.180 ± 0.217	0.948 ± 0.034
RankCosine-sqrt	0.080 ± 0.159	0.886 ± 0.056
RankCosine-l	0.917 ± 0.112	0.999 ± 0.002
RankCosine-q	0.102 ± 0.161	0.890 ± 0.060
RankCosine-exp	0.047 ± 0.163	0.746 ± 0.136

Experimental Results on OHSUMED



Conclusion and Future Work

- Study has been made on the listwise approach to learning to rank.
  - Likelihood loss seems to be the best listwise loss functions under investigation, according to both theoretical and empirical studies.

- Furthermore
  - In addition to consistency, rate of convergence and generalization ability should also be studied.
  - In real ranking problems, the true loss should be cost-sensitive (e.g. NDCG in Information Retrieval).

Thanks!

tyliu@microsoft.com

http://research.microsoft.com/users/tyliu/