Supplementary Information for "Structured Ranking Learning using Cumulative Distribution Networks"

Jim C. Huang
Probabilistic and Statistical Inference Group
University of Toronto
Toronto, ON, Canada M5S 3G4
jim@psi.toronto.edu

Brendan J. Frey
Probabilistic and Statistical Inference Group
University of Toronto
Toronto, ON, Canada M5S 3G4
frey@psi.toronto.edu

1 Computing gradients for structured ranking learning

Given observations $\mathcal{D} = \{D_1, \cdots, D_N\}$, the problem of structured ranking learning is given by

$$\inf_{\boldsymbol{\theta}} \quad \sum_t \sum_{e,e'} \log\left(1 + \exp\big(-w_1 r_e(\mathbf{a}; D_t)\big) + \exp\big(-w_2 r_{e'}(\mathbf{a}; D_t)\big)\right) \quad \text{s.t.} \quad \boldsymbol{\theta} \geq 0$$

$$\|\boldsymbol{\theta}\|_1 \leq t. \quad (1)$$

For the observation D_t, the gradient $\nabla_{\mathbf{a}}\mathcal{L}(\boldsymbol{\theta}; D_t)$ with respect to the loss $\mathcal{L}(\boldsymbol{\theta}; D_t)$ for that observation is given by

$$\nabla_{\mathbf{a}}\mathcal{L}(\boldsymbol{\theta}; D_t) = -\sum_{e,e'} \frac{1}{\phi(r_e, r_{e'})}\Big(\partial_{r_e}\Big[\phi(r_e, r_{e'})\Big]\nabla_{\mathbf{a}} r_e(\mathbf{a}; D_t) + \partial_{r_{e'}}\Big[\phi(r_e, r_{e'})\Big]\nabla_{\mathbf{a}} r_{e'}(\mathbf{a}; D_t)\Big),$$

with

$$\begin{aligned}
\partial_{r_e}\Big[\phi(r_e, r_{e'})\Big] &= -w_1 \exp(-w_1 r_e)\ \phi(r_e, r_{e'}) \\
\partial_{r_{e'}}\Big[\phi(r_e, r_{e'})\Big] &= -w_2 \exp(-w_2 r_{e'})\ \phi(r_e, r_{e'}) \\
\nabla_{\mathbf{a}} r_e(\mathbf{a}; D_t) &= \nabla_{\mathbf{a}}\rho(\mathbf{x}_i; \mathbf{a}) - \nabla_{\mathbf{a}}\rho(\mathbf{x}_j; \mathbf{a}) \\
\nabla_{\mathbf{a}}\rho(\mathbf{x}; \mathbf{a}) &= \mathbf{a}\frac{\sum_i \big(y_i - \rho(\mathbf{x}; \mathbf{a})\big)\|\mathbf{x} - \mathbf{x}_i\|^2 K(\mathbf{x}_i, \mathbf{x}; \mathbf{a})}{\sum_i K(\mathbf{x}_i, \mathbf{x}; \mathbf{a})}.
\end{aligned}$$

The derivatives with respect to the CDN function weights w_1, w_2 are given by

$$\begin{aligned}
\partial_{w_1}\Big[\mathcal{L}(\boldsymbol{\theta}; D_t)\Big] &= -\sum_{e,e'} r_e \exp\big(-w_1 r_e\big)\ \phi(r_e, r_{e'}) \\
\partial_{w_2}\Big[\mathcal{L}(\boldsymbol{\theta}; D_t)\Big] &= -\sum_{e,e'} r_{e'} \exp\big(-w_2 r_{e'}\big)\ \phi(r_e, r_{e'}).
\end{aligned}$$

2 Supplementary Results

In addition to the OHSUMED dataset, we also applied the structured ranking learning framework to the ".gov" collection of the TREC2004 web track, provided as part of the LETOR 2.0 benchmarks [4]. This dataset consists of a total of 75 queries of 1000 documents each, with 44 features per query-document pair. The relevance labels used in the TREC2004 dataset are the same as those for the OHSUMED dataset, namely *definitely relevant*, *partially relevant* or *not relevant*. Using the same training and model selection procedure described in the main paper, we computed the Precision, MAP and NDCG performance metrics of our method: this is shown in Figures 1(a),1(b),1(c) in addition to the performances of six other ranking methods which are provided as part of the LETOR 2.0 benchmarks.

3 RankNet and ListNet/ListMLE as CDNs with particular graph topologies

The RankNet and ListNet/ListMLE probability models for learning to rank [1, 2] can be viewed as disconnected and partially connected CDNs respectively. In the case of RankNet, the resulting CDN model is optimized using cross-entropy loss. The ListNet/ListMLE models are instances of Plackett-Luce models [5] in which preferences between objects to be ranked are partially connected in the corresponding CDN. An example demonstrating these models as CDNs is shown in Figure 2 for a toy example involving 4 nodes $V_1 \succ V_2 \succ V_3 \succ V_4$ with preference variables $\pi_{12}, \pi_{13}, \pi_{14}, \pi_{23}, \pi_{24}, \pi_{34}$.

References

[1] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender. Learning to rank using gradient descent. *In Proceedings of the Twenty-Second International Conference on Machine Learning (ICML)*, 2005.

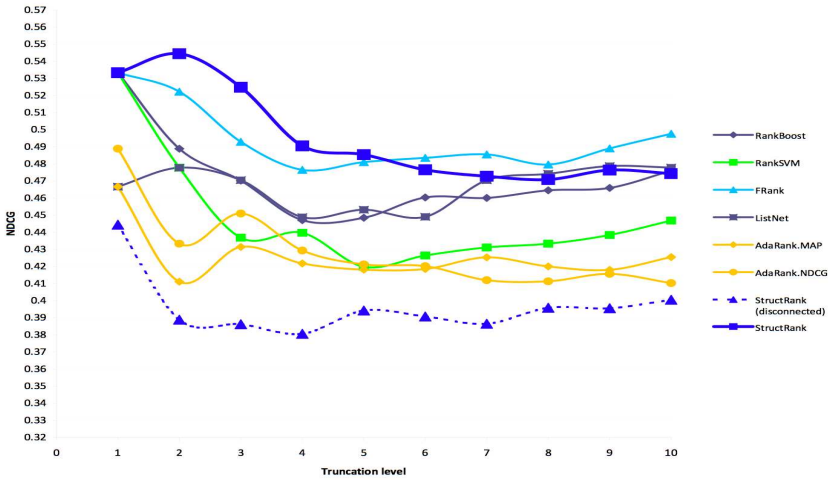
[2] Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai and H. Li. Learning to rank: from pairwise approach to listwise approach. *In Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML)*, 2007.

[3] J.C. Huang and B.J. Frey. Cumulative distribution networks and the derivative-sum-product algorithm. *In Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

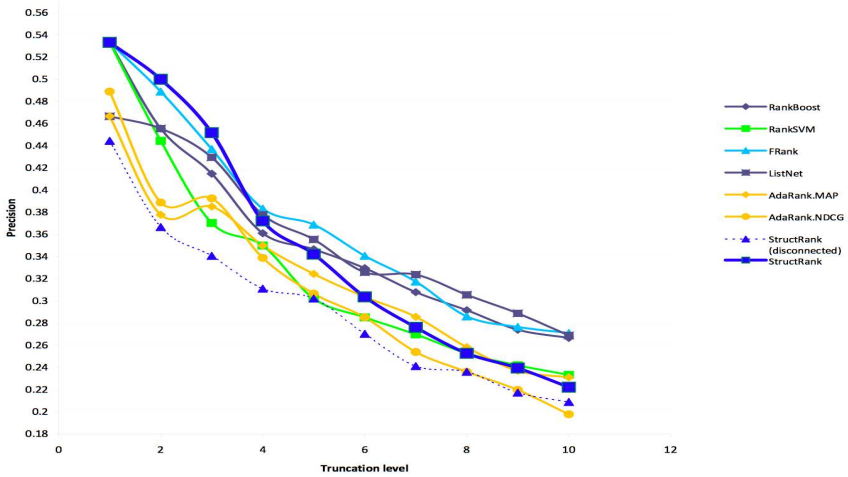
[4] T.Y. Liu, J. Xu, T. Qin, W. Xiong and H. Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. *LR4IR 2007, in conjunction with SIGIR 2007*, 2007.

[5] J. I. Marden. Analyzing and modeling rank data. *CRC Press*, 1995.

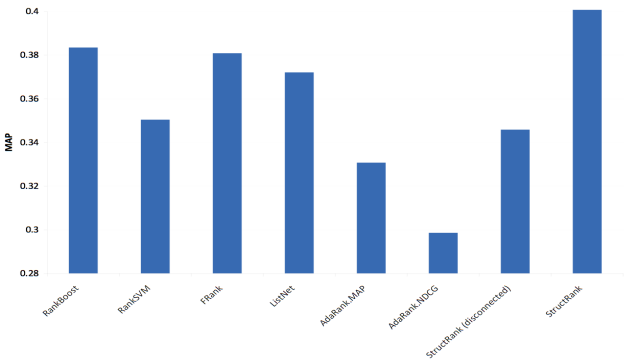
[6] F. Xia, T.Y. Liu, J. Wang, W. Zhang and H. Li. Listwise approach to learning to rank - theory and algorithm. *In Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, 2008.



(a)

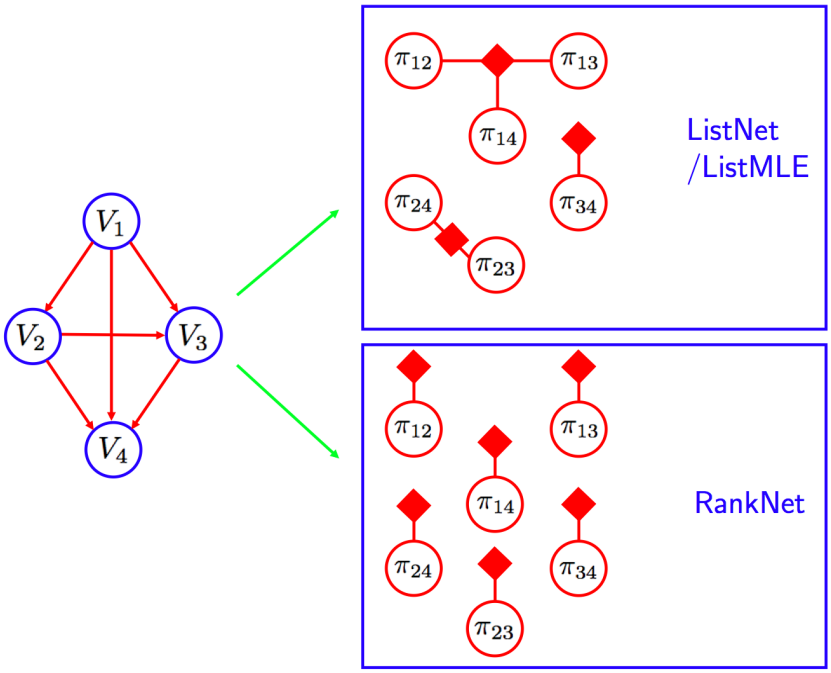


(b)



(c)

Supplementary Figure 1: Results on the TREC 2004 dataset of the LETOR benchmark. a) Average NDCG as a function of truncation level n for the TREC2004 dataset. NDCG values are averaged over 5 cross-validation splits; b) Mean average precision (MAP) as a function of truncation level n; c) Mean average precision value for several methods.



Supplementary Figure 2: The ListNet/ListMLE and RankNet probabilistic models represented as CDNs with particular topologies for an example order graph representing the ordering $V_1 \succ V_2 \succ V_3 \succ V_4$. In the case of RankNet, the corresponding CDN is disconnected, as preference variables are assumed to be independent. The ListNet/ListMLE model is an example of a Plackett-Luce model[5].