

SoftRank

Optimising Non-Smooth Rank Metrics

Mike Taylor, John Guiver,
Steve Robertson and Tom Minka

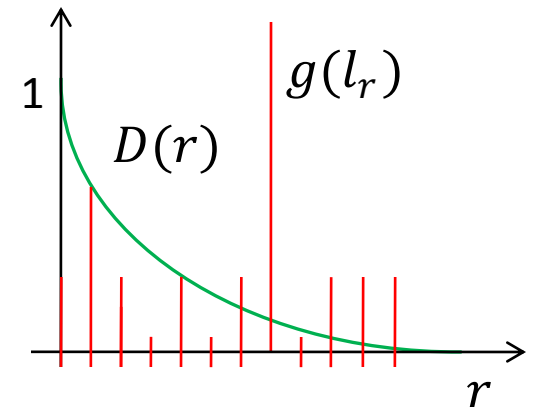
Microsoft Research Cambridge

IR Metrics are Not Smooth: NDCG

- Document score $s = f(\mathbf{x}, \mathbf{w})$
 - learn \mathbf{w} from labelled training data
- *Sort* gives ranked list of documents
 - With labels $l \in \{0,1,2,3\}$
 - And gains $g(l) = 2^l$

- Normalized DCG:

$$G = G_{max}^{-1} \sum_{r=0}^{N-1} g(l_r) D(r)$$



- $\frac{\partial G}{\partial \mathbf{w}}$ non-smooth: generally zero, but...
 - Infinite as documents switch ranks
 - Makes gradient-based optimisation tricky

Previous Proxy Training Objectives

- Pointwise:

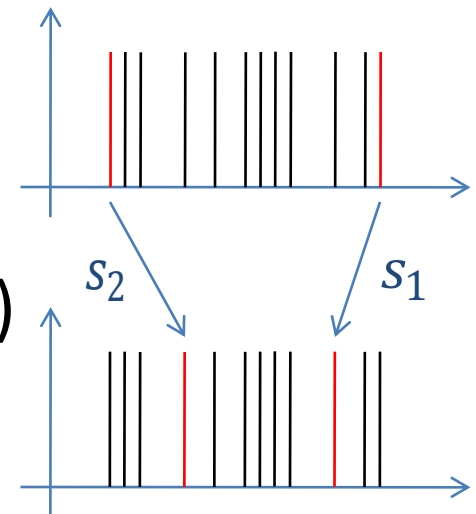
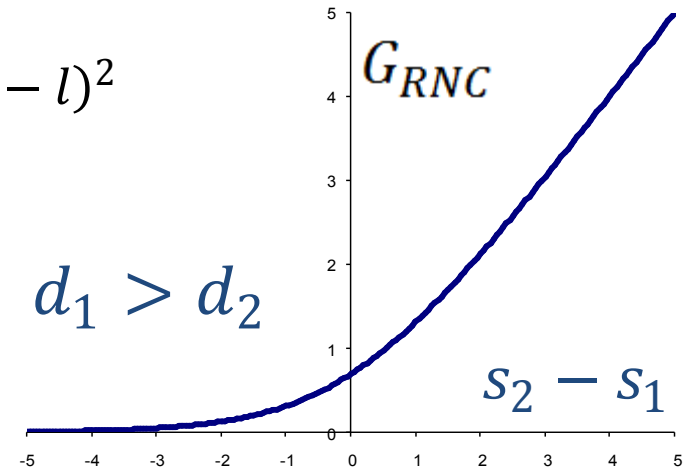
- Regression on labels $G_{MSE} = (s - l)^2$
- Ordinal regression

- Pairwise:

- RankSvm
- RankNet $G_{RNC} = \log(1 + e^{s_2 - s_1})$

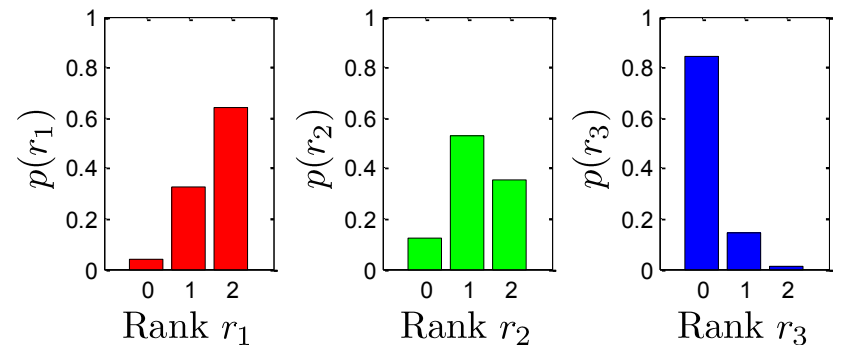
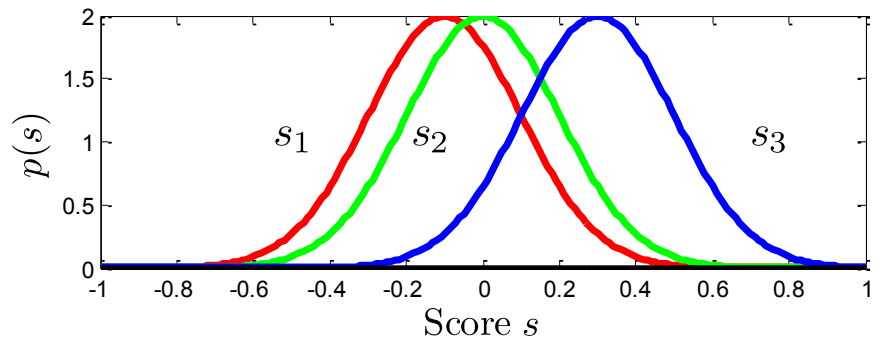
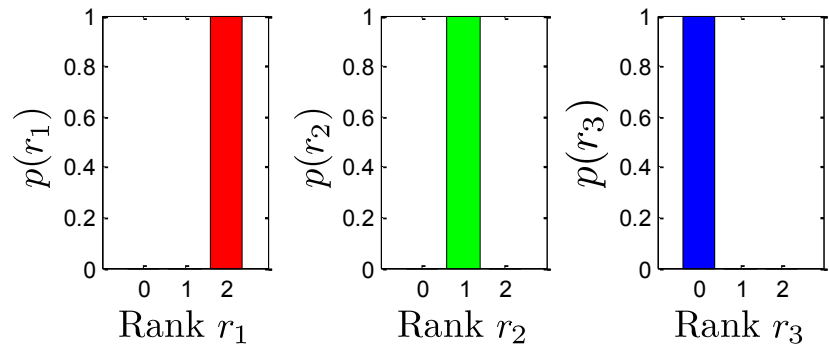
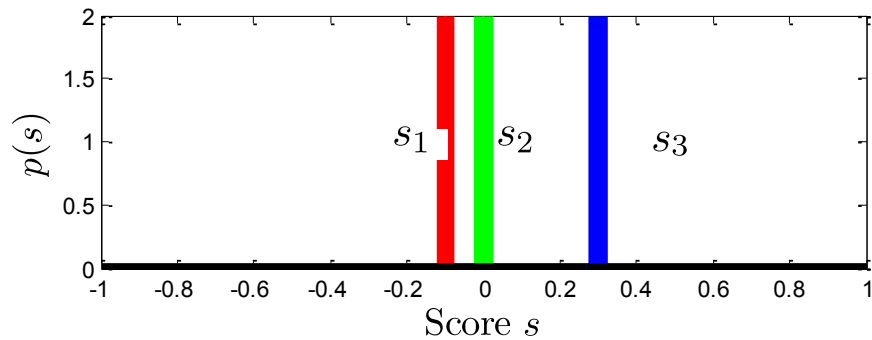
- List based:

- LambdaRank $\left| \frac{\partial G_{LR}}{\partial s_1} \right| \gg \left| \frac{\partial G_{LR}}{\partial s_2} \right|$
- Yue (Structural SVM), Cao (ListNet)



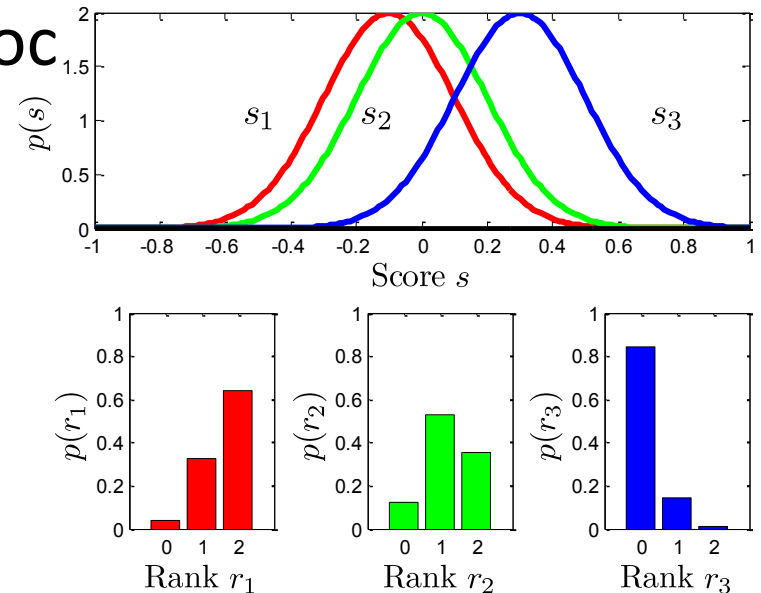
SoftRank Approach

- Add noise to scores
- Infer rank distribution for each doc – no sort
- SoftNDCG is $E[\text{NDCG}]$ under rank distribution
- Derivatives of SoftNDCG are smooth



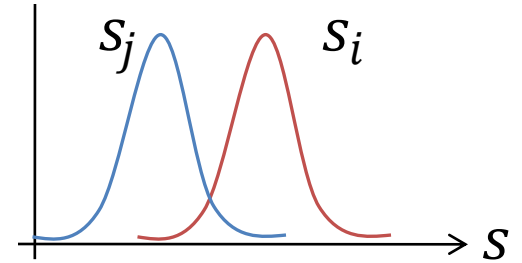
Rank Distribution

- Generative process for exact distribution:
 - Sample from each score Gaussian
 - *Sort* N samples to get rank for each doc
 - Accumulate ranks for each doc
- Doubly stochastic matrix R
 - Rank distribution given doc
 - Doc distribution given rank
- Need a good approximation with no sort

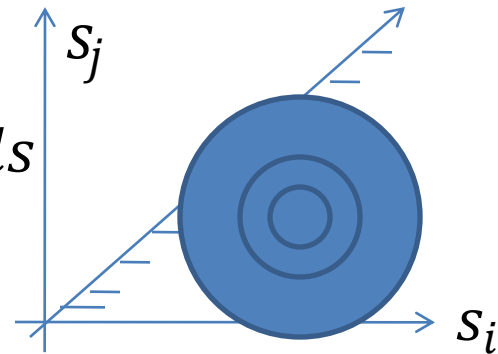


Pairwise Contest Approximation

- Assume scores are Gaussian
 - NN Ranking function gives means
- Pair-wise contest:



$$\pi_{ij} \equiv Pr(s_i - s_j > 0) = \int_0^{\infty} \mathcal{N}(s | \bar{s}_i - \bar{s}_j, 2\sigma^2) ds$$



- Expected Rank = num times beaten:

$$E[r_j] = \sum_{i=1, i \neq j}^N \pi_{ij} \quad \text{No sort so differentiable}$$

- Rank-Binomial : Sum $N - 1$ Bernoulli(π_{ij}) trials

Rank Distribution Recursion

$$\pi_{ij} \equiv \Pr(s_i > s_j)$$

$$p_j^1(r) = \delta(r)$$

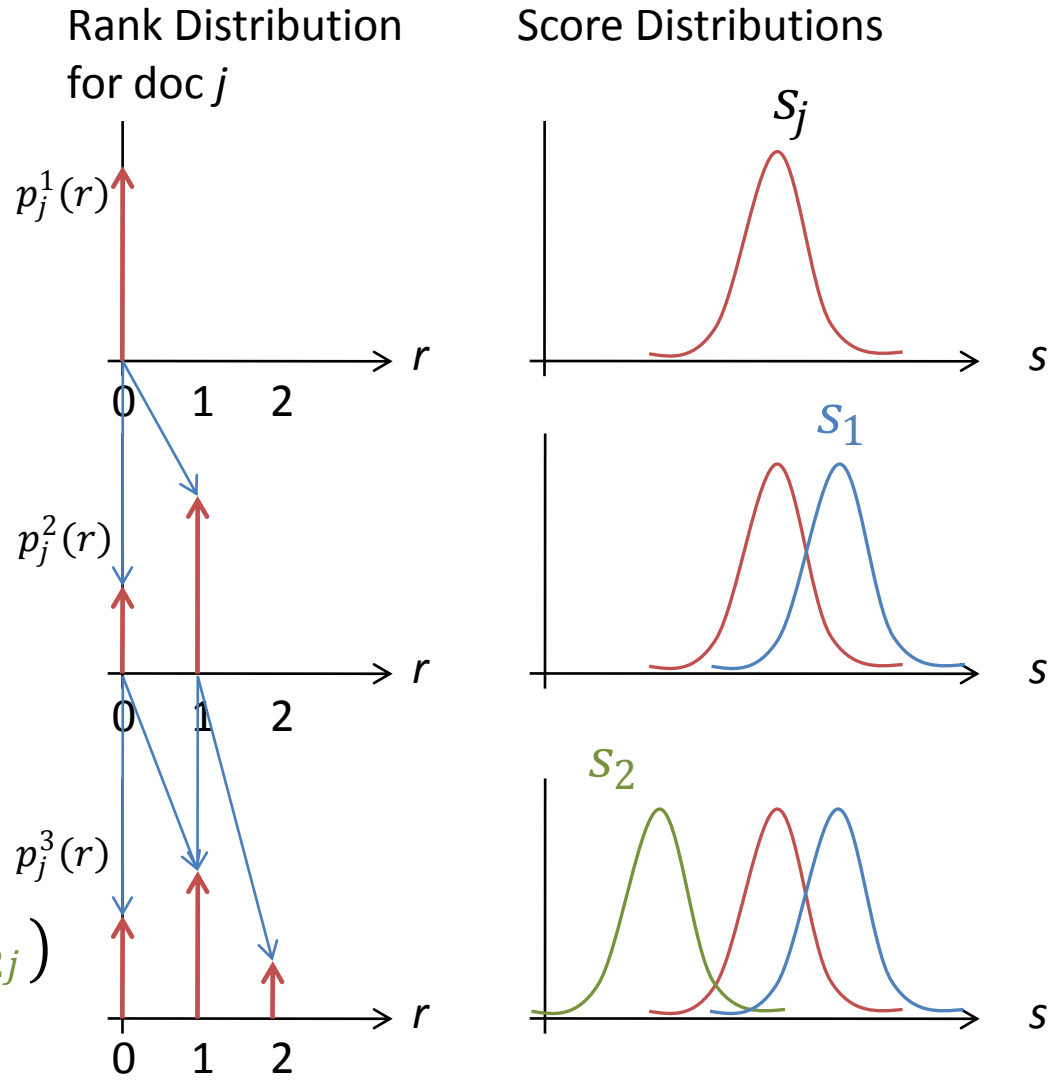
$$p_j^2(0) = 1 - \pi_{1j}$$

$$p_j^2(1) = \pi_{1j}$$

$$p_j^3(1) = p_j^2(0)\pi_{2j} + p_j^{i-1}(1)(1 - \pi_{2j})$$

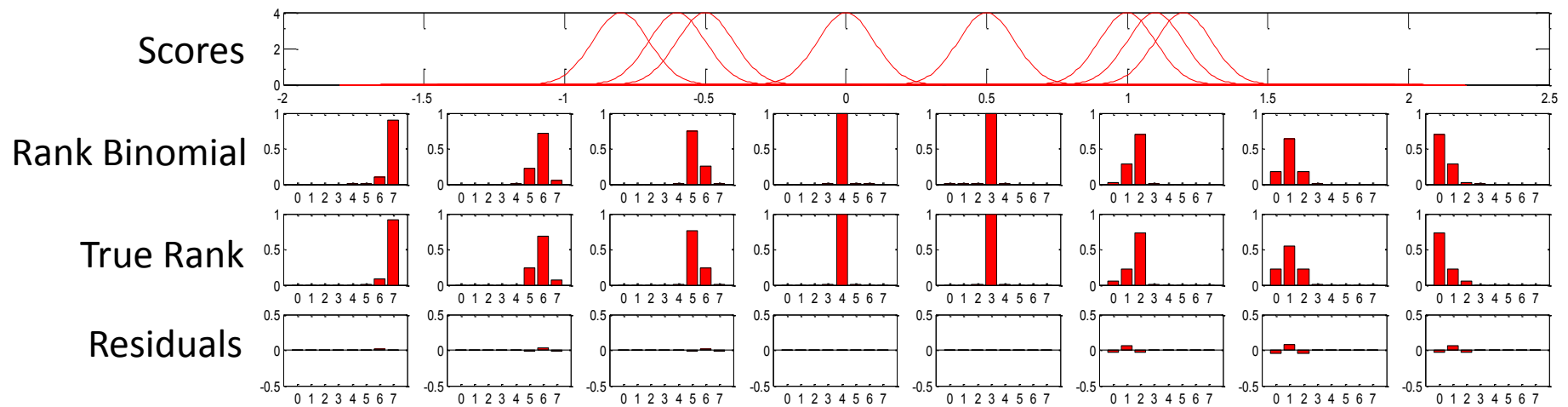
Take home:

- Gives expressions for $p_j(r)$ and $\frac{\partial p_j(r)}{\partial \bar{s}}$ analytic in score means



Effect of Pairwise Contest Trick

- Rank-Binomial: no sort but an approximation
- Compare with exact distribution



- Qualitatively a good approximation
- Can improve with row/col normalizations

SoftNDCG

- NDCG: $G = G_{max}^{-1} \sum_{j=1}^N g(l_j) D(r_j)$

- SoftNDCG: $\mathcal{G} \equiv G_{max}^{-1} \sum_{j=1}^N g(l_j) E[D(r_j)]$

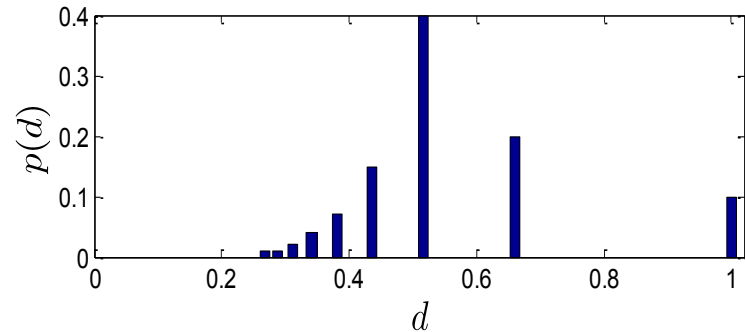
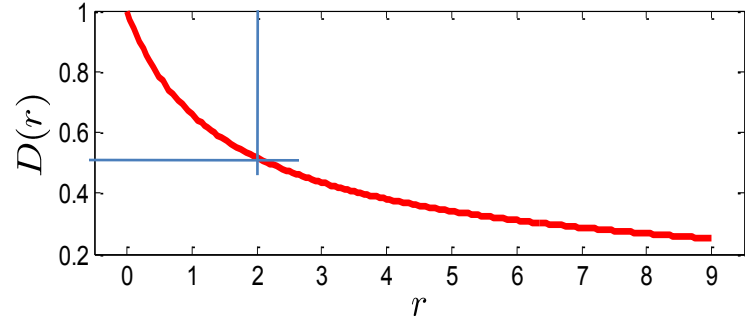
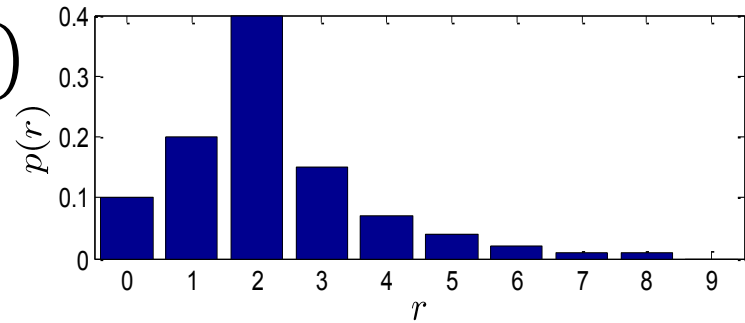
- And so

$$\mathcal{G} = G_{max}^{-1} \sum_{j=1}^N g(l_j) \sum_{r=0}^{N-1} D(r) p_j(r)$$

Using $\frac{\partial p_j(r)}{\partial \bar{s}}$ from rank recursion

From backprop

- Seek: $\frac{\partial \mathcal{G}}{\partial \mathbf{w}} = \frac{\partial \mathcal{G}}{\partial \bar{s}} \frac{\partial \bar{s}}{\partial \mathbf{w}}$



Experiments

- 2-layer neural net
 - Web: 300 features, Train 4K, Val/Test 2K each
- NDCG@10 used for validation/test
- Stochastic gradient descent with restarts
 - Quite sensitive to learning rate and smoothing σ
 - Set using validation set

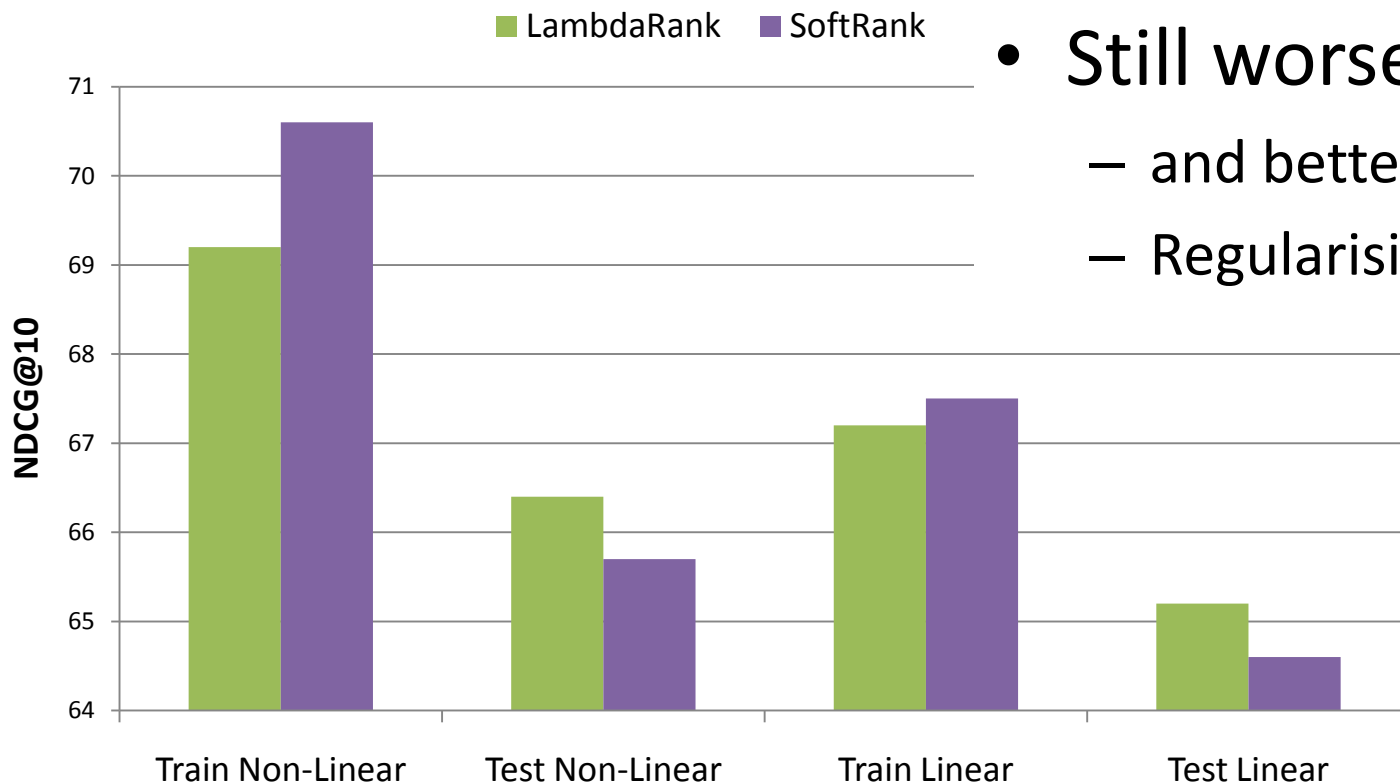
SoftRank Results



- Better training set NDCG
 - Optimisation is better: approximations are good
- Web: Worse than LambdaRank on test set
 - Somehow not generalising

Generalization Study

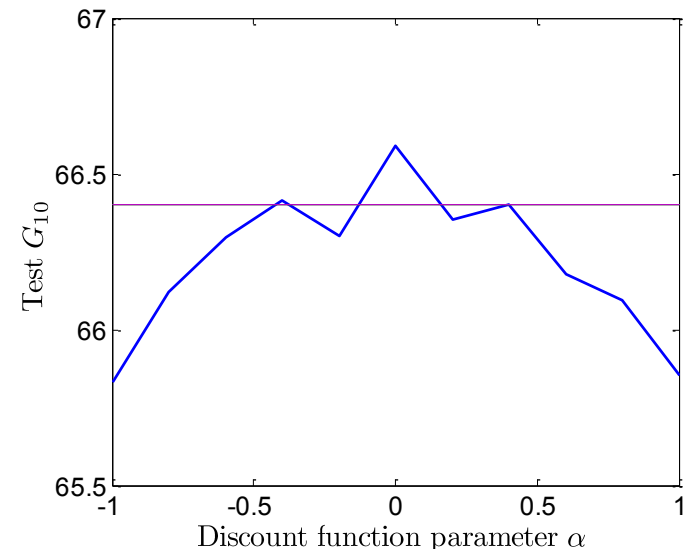
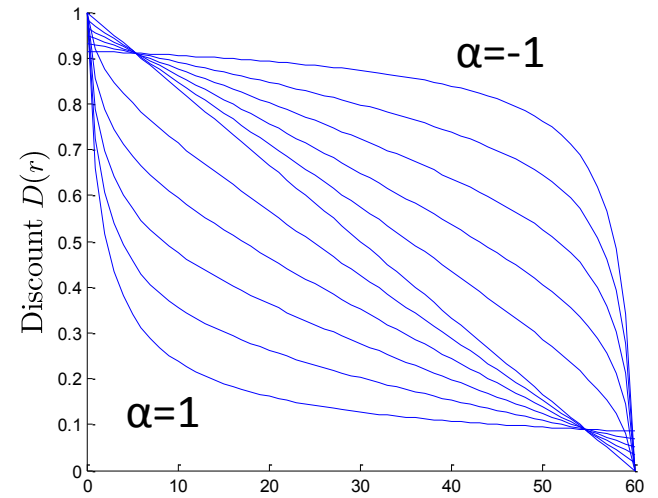
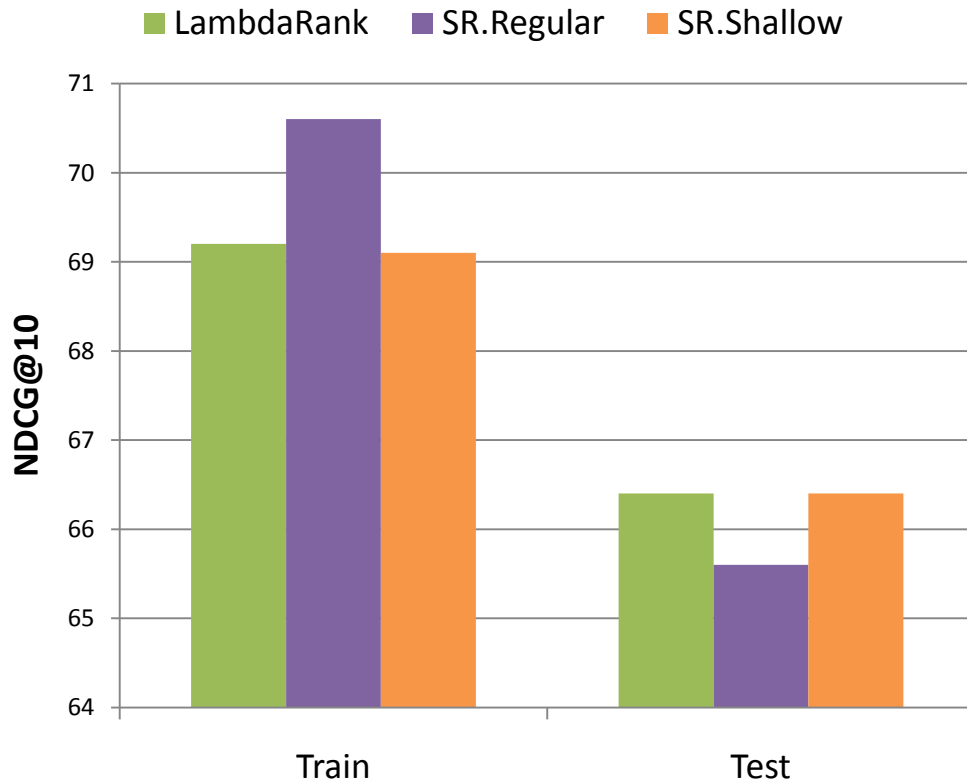
- Are we somehow just overfitting?
 - Despite NDCG validation
 - Try much simpler linear model (3K \rightarrow 300 parameters)



- Still worse on test set
 - and better training set
 - Regularising OK

Discount Study

- SoftRank focuses *too* much on top ranks?
 - Useful info in ordering at lower ranks
 - Effectively ignored by SoftRank
 - Try shallower discounts



Conclusions

- SoftRank optimises NDCG very well
- Can use Rank-Binomial to optimise other rank-based IR metrics
- SoftNDCG does not generalize well
 - Seems to focus *too much* on top ranks in training
 - Inefficient use of training data
 - Training objective *should* be different
- Can recover LambdaRank performance with less severe discount function