

# Exponential Family Graph Matching and Ranking

Authors

James Petterson, Tibério S. Caetano, Julian J. McAuley and Jin Yu

SML, NICTA & RISE, ANU

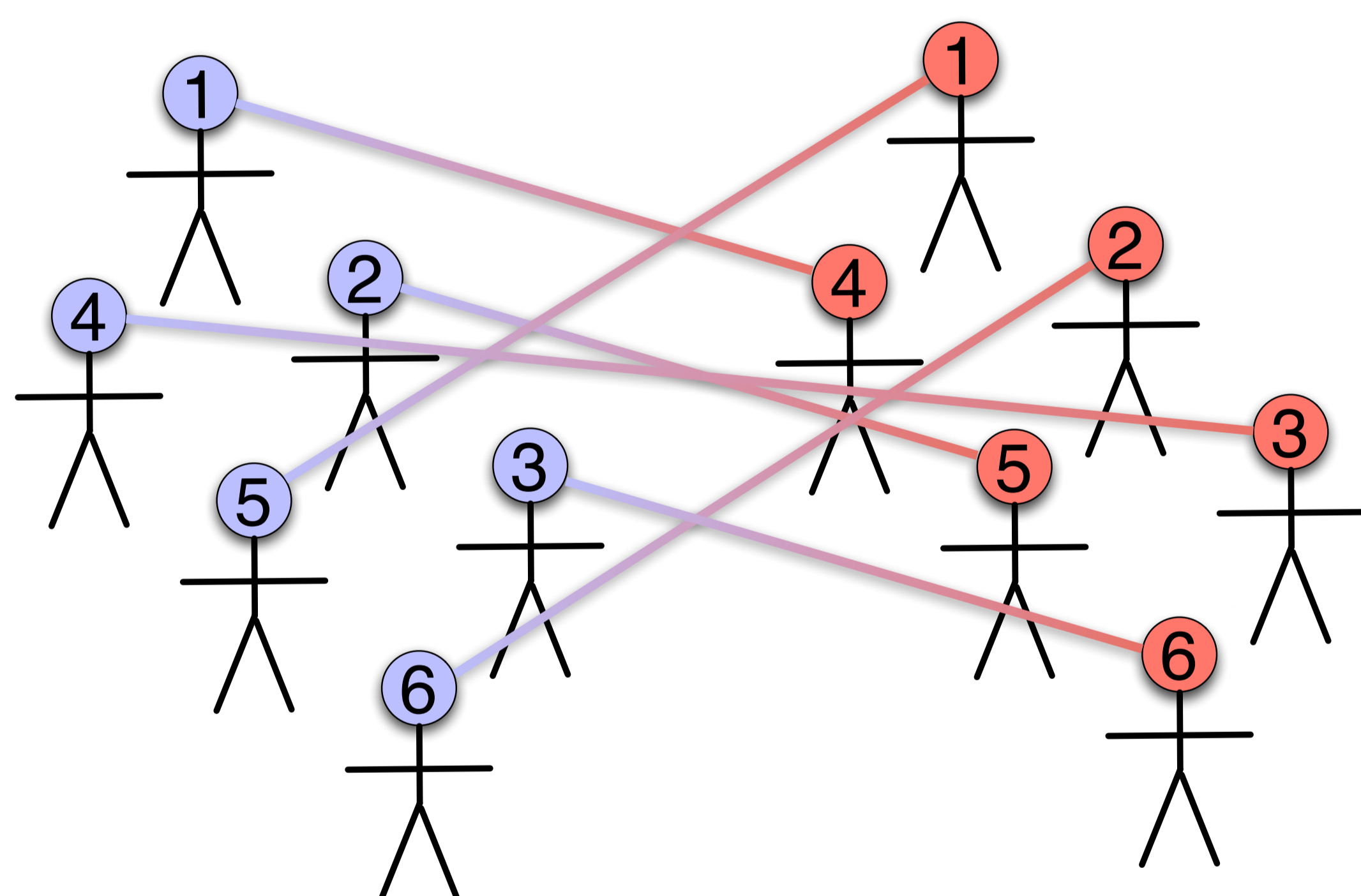
e-mails: [first.last@nicta.com.au](mailto:first.last@nicta.com.au)

source code at <http://users.rise.anu.edu.au/~jpetterson/>

## Abstract

- We present a method for **learning** max-weight matching predictors in bipartite graphs.
- The method consists of performing **maximum a posteriori** estimation in **exponential families** with sufficient statistics that encode permutations and data features.
- Although inference is in general hard, we show that for one very relevant application – **document ranking** – exact inference is efficient. For general model instances, an appropriate **sampler** is readily available.
- Contrary to existing max-margin matching models, our approach is **statistically consistent** and, in addition, experiments with increasing sample sizes indicate superior **improvement** over such models.

## The Problem



Each couple  $ij$  has a pairwise happiness score  $w_{ij}$ .

Monogamy is enforced, and no person can be unmatched.

Goal is to maximize **overall** happiness, i.e.:

$$y^* = \operatorname{argmax}_y \sum_{i=1}^m w_{iy(i)}$$

where  $y$  is a permutation. This is a well-studied problem; it is tractable and can be solved in  $O(m^3)$  time (Papadimitriou and Steiglitz, 1982).

## The Model

We relax the assumption that we know the scores  $w_{ij}$ , since in reality what we measure are the edge features  $x_{ij} = (x_{ij}^1, \dots, x_{ij}^d)$  of dimension  $d$ .

Therefore, we parameterize them:  $w_{ij} = \langle x_{ij}, \theta \rangle$

and perform MAP estimation of the parameters.

We assume an exponential family model, where the probability model is

$$p(y|x; \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(x; \theta)), \text{ where}$$

$$g(x; \theta) = \log \sum_y \exp \langle \phi(x, y), \theta \rangle$$

is the log-partition function and  $y$  is a permutation.

We impose a Gaussian prior on  $\theta$ . We minimize the negative log-posterior  $\ell(Y|X; \theta)$ , which becomes our loss function:

$$\ell(Y|X; \theta) = \frac{\lambda}{2} \|\theta\|^2 + \frac{1}{N} \sum_{n=1}^N (g(x^n; \theta) - \langle \phi(x^n, y^n), \theta \rangle)$$

where  $\lambda$  is a regularization constant.

## Feature Parameterization

The key point is that we equate the solution of the matching problem to the prediction of the exponential family model, i.e.,  $\sum_i w_{iy(i)} = \langle \phi(x, y), \theta \rangle$ . Since our goal is to parameterize features of individual pairs of nodes (so as to produce the weight of an edge), the most natural model is

$$\phi(x, y) = \sum_{i=1}^M x_{iy(i)}, \text{ which gives}$$

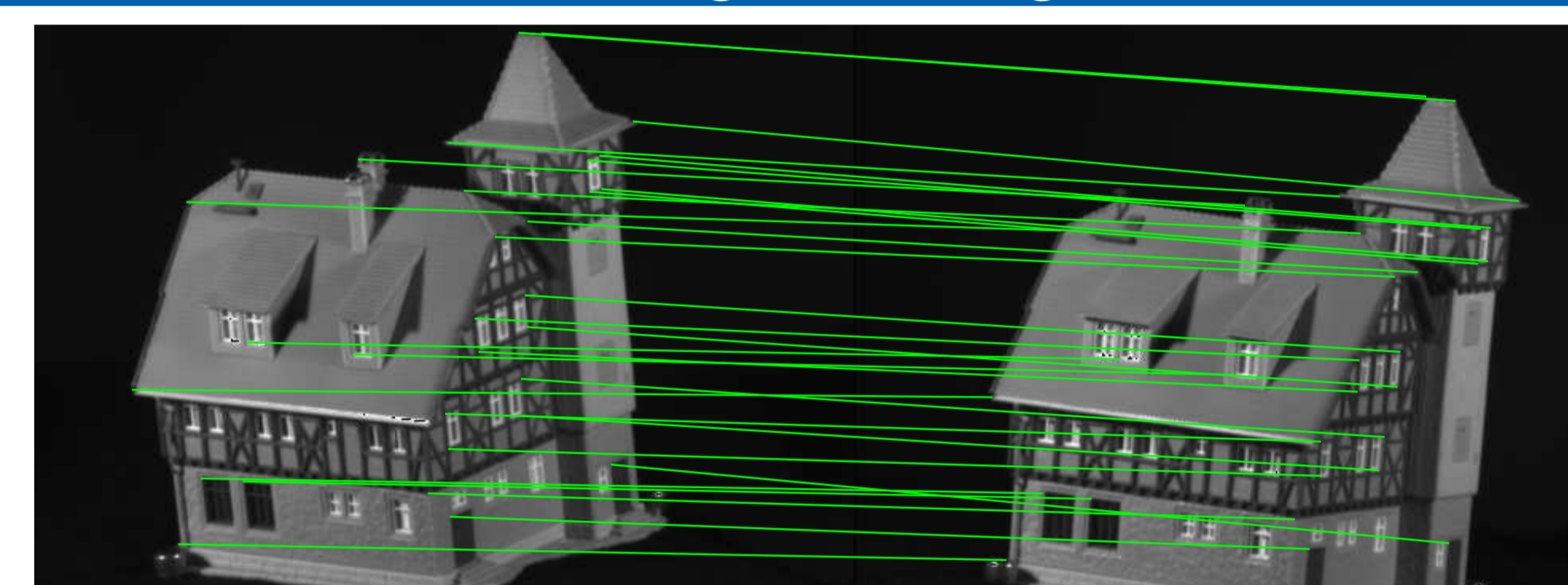
$$w_{iy(i)} = \langle x_{iy(i)}, \theta \rangle,$$

## Learning

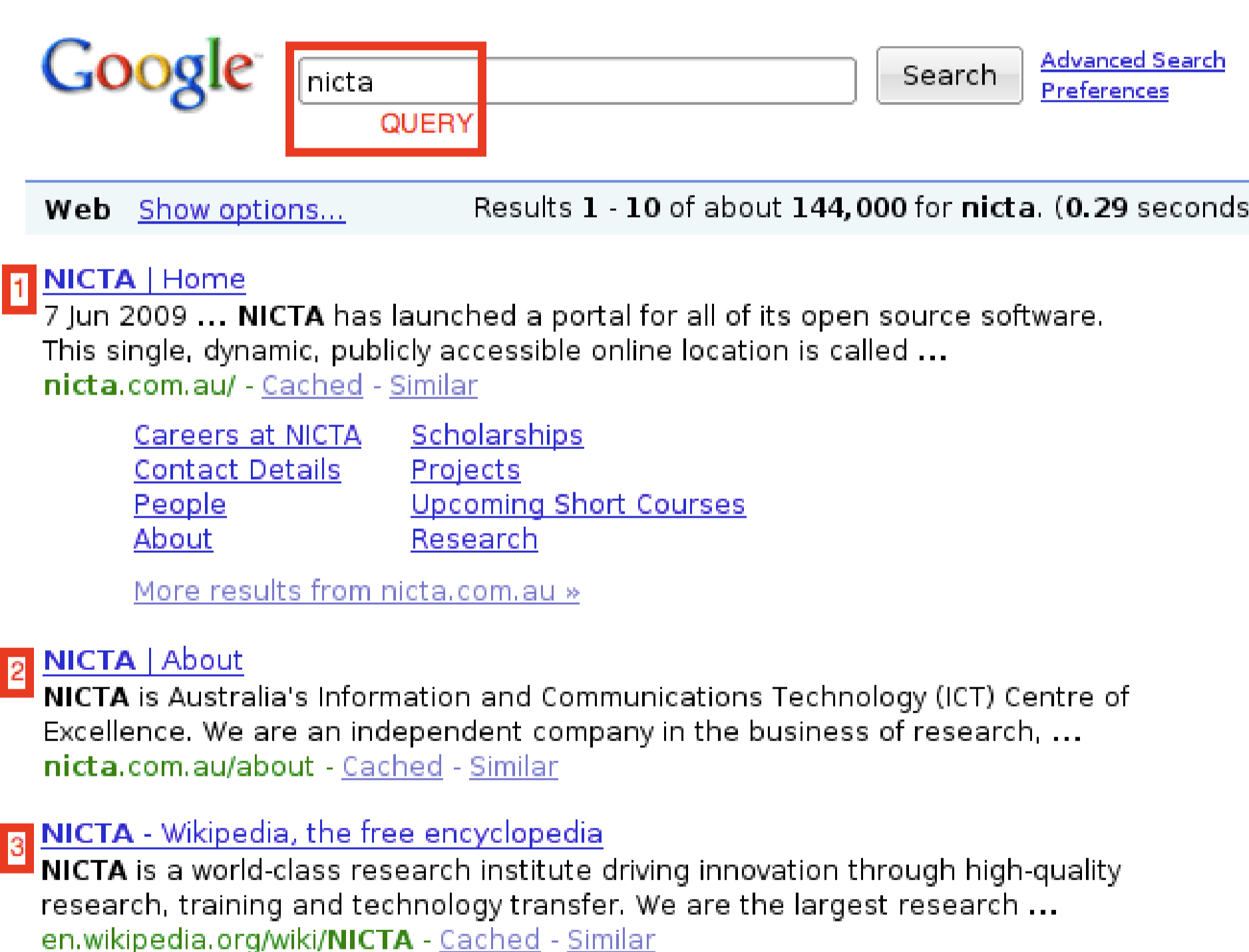
- Since our loss is a convex and differentiable function of  $\theta$  gradient descent will find the **global optimum**.
- The only issue is the computation of the gradient of the partition function, which involves the computation of the permanent of a matrix, a #P-complete problem.
- For small problems (e.g. document ranking) this can be done exactly, but for larger problems (e.g. image matching) we need to resort to an approximation – the algorithm that Huber and Law recently proposed (SODA, 2008), which produces **exact samples** from the distribution of perfect matches on weighted bipartite graphs.

## Applications

### Image Matching



### Document Ranking



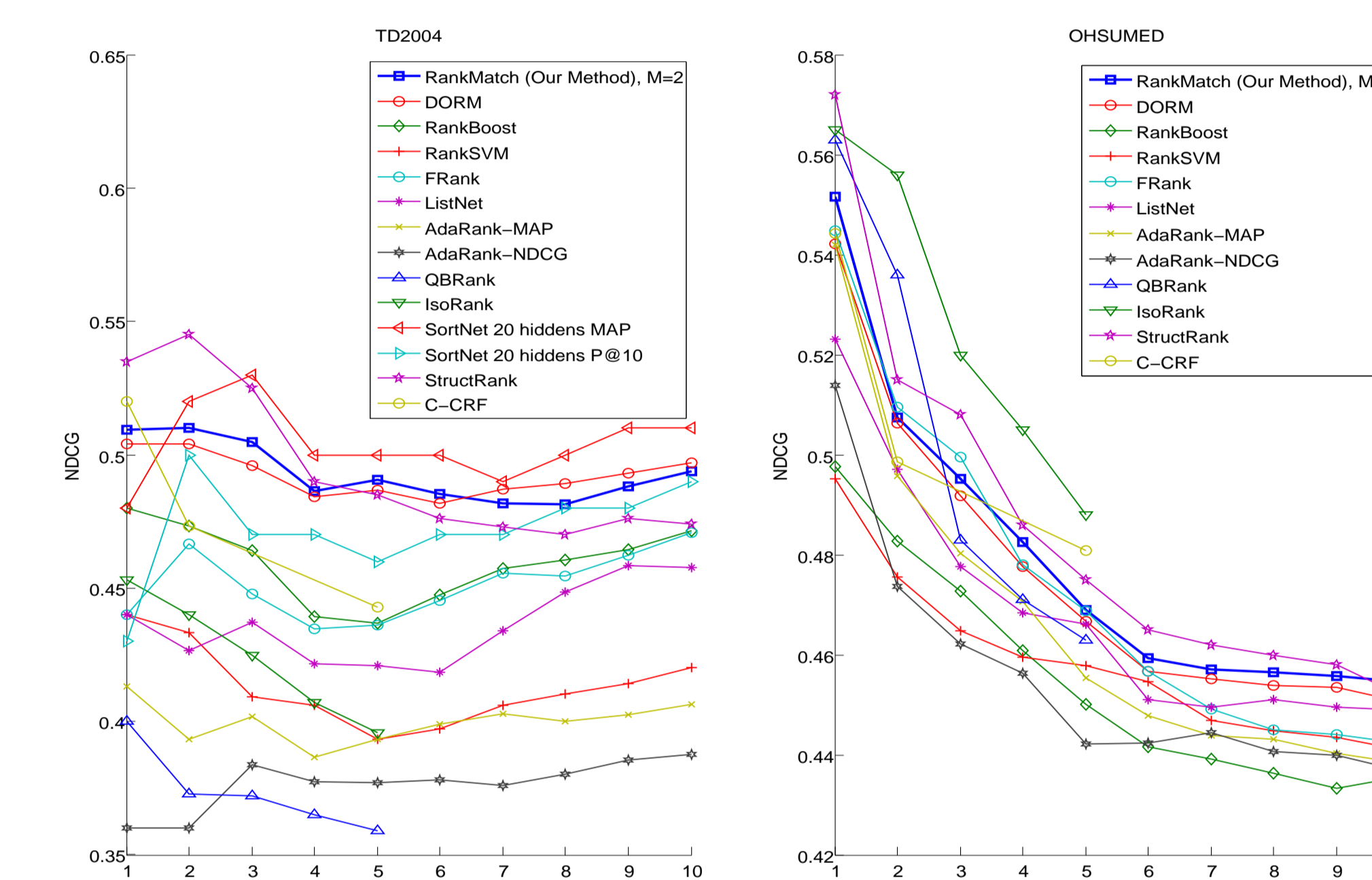
## Comparison to Max-Margin

Compared to existing max-margin approaches (Le and Smola, 2007), our model has some advantages:

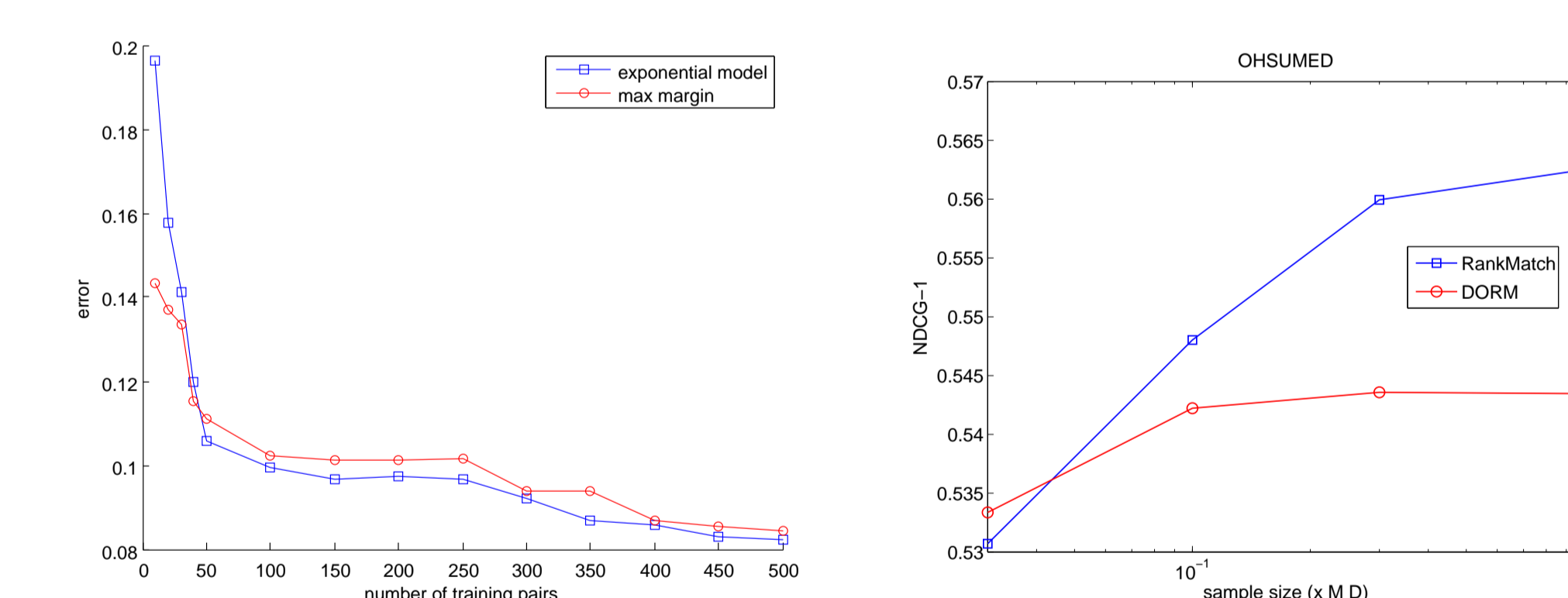
- it is **consistent** – that is, in the limit of infinite training data it will obtain the best attainable model.
- it is a **probabilistic model**, and therefore can be integrated as a module in a Bayesian framework.
- it is **simpler**.

The main drawback is that, except for small graphs, it can be considerably **slower**.

## Results



Ranking application: our method (RankMatch) gets state-of-the-art results in the LETOR 2.0 learning to rank dataset. Note that in this particular application **inference** can be done in **linear time**.



Left: classification error in a graph matching application. Right: NDCG-1 score in a ranking application. Compared to the max-margin approach, our method shows better results as more training data is added.

## References

- Caetano, T. S., McAuley, J., Cheng, L., Le, Q. V. & Smola, A. J. (2009). *Learning graph matching*. IEEE Trans. on PAMI, 31, 1048–1058.
- Huber, M. & Law, J. (2008). *Fast approximation of the permanent for very dense problems*. SODA.
- Le, Q. & Smola, A. (2007). *Direct optimization of ranking measures*. <http://arxiv.org/abs/0704.3359>.
- Liu, T.-Y., Xu, J., Qin, T., Xiong, W. & Li, H. (2007). *Letor: Benchmark dataset for research on learning to rank for information retrieval*. LR4IR.
- McAllester, D. (2007). *Generalization bounds and consistency for structured labeling*. Predicting Structured Data.