Exploiting Within-Clique Factorizations in Junction-Tree Algorithms

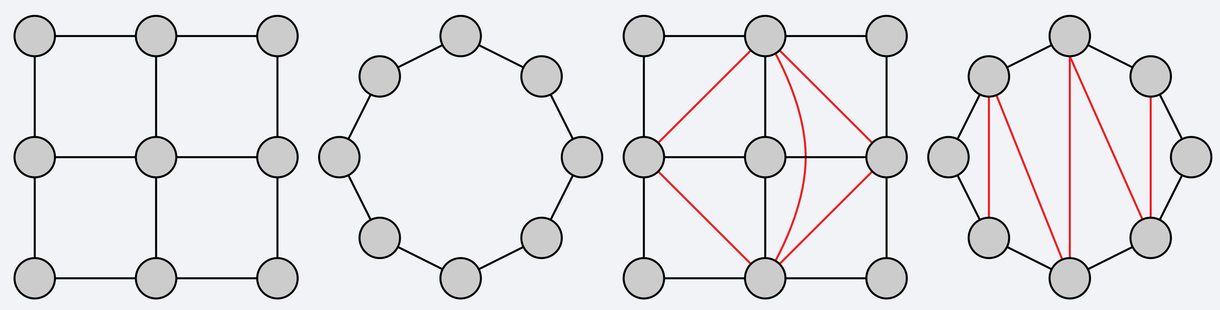
Julian McAuley, Tibério Caetano

Australian National University/NICTA

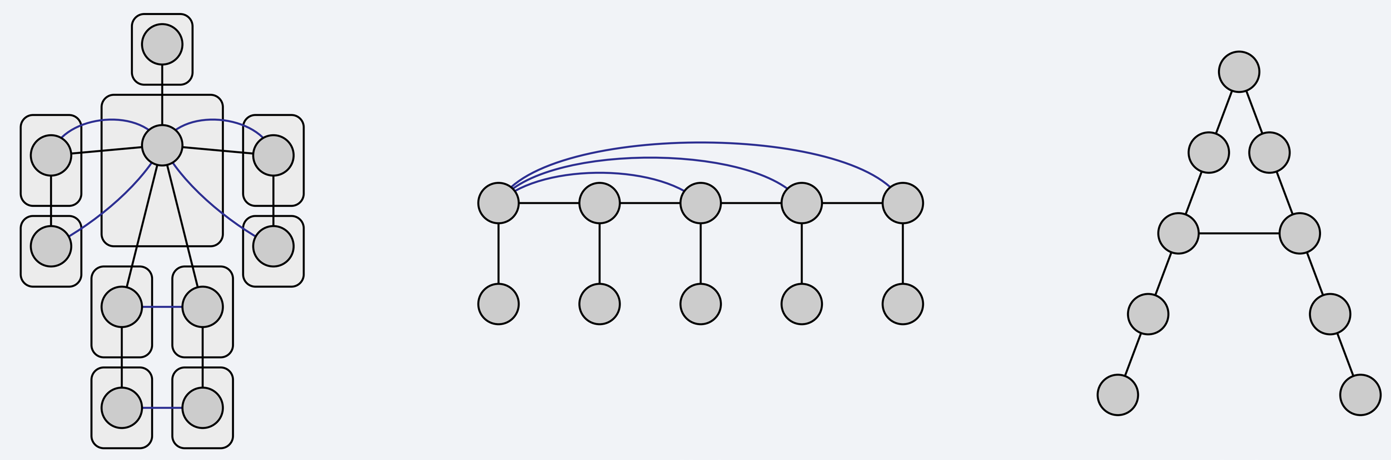
Abstract

We show that the expected computational complexity of the Junction-Tree Algorithm for *maximum a posteriori* inference in graphical models **can be improved**. Our results apply whenever the potentials over maximal cliques of the triangulated graph are factored over subcliques. This enlarges the class of models for which exact inference is efficient.

Examples of graphs whose potentials factorize



The graphical models shown above contain only pairwise factors; triangulating them increases their maximal clique size.



(a) (b) (c)

Analogous cases are common in many applications: (a) a model for pose reconstruction from [1]; (b) a 'skip-chain CRF' from [2]; (c) a model for deformable matching from [3]. Although the (triangulated) models have cliques of size three, they **factorize** into pairwise terms.

The fundamental step in MAP-estimation

In order to pass messages and compute maximum-likelihood states in graphical models we need to find the index that chooses the maximum product amongst two lists:

$$\hat{i} = \underset{i \in \{1 \ldots N\}}{\operatorname{argmax}} \{\mathbf{v}_a[i] \times \mathbf{v}_b[i]\}.$$

Although this seems to be a **linear** time operation, it can be reduced to $O(\sqrt{N})$ (in the expected case) if we know the permutations that sort $\mathbf{v}_a$ and $\mathbf{v}_b$. Our results arise due to the fact that knowing these permutations allows us to ignore much of the search space:

value	99	92	87	81	78	66	53	46	30	26	21	16	12	10	8	6
index before sorting	6	2	14	16	9	7	12	8	10	3	11	13	1	15	4	5

we don't need to search behind this line

index before sorting	3	4	8	11	7	16	13	9	6	2	15	10	12	5	1	14
value	98	93	85	76	71	70	67	65	63	57	48	42	39	37	26	17

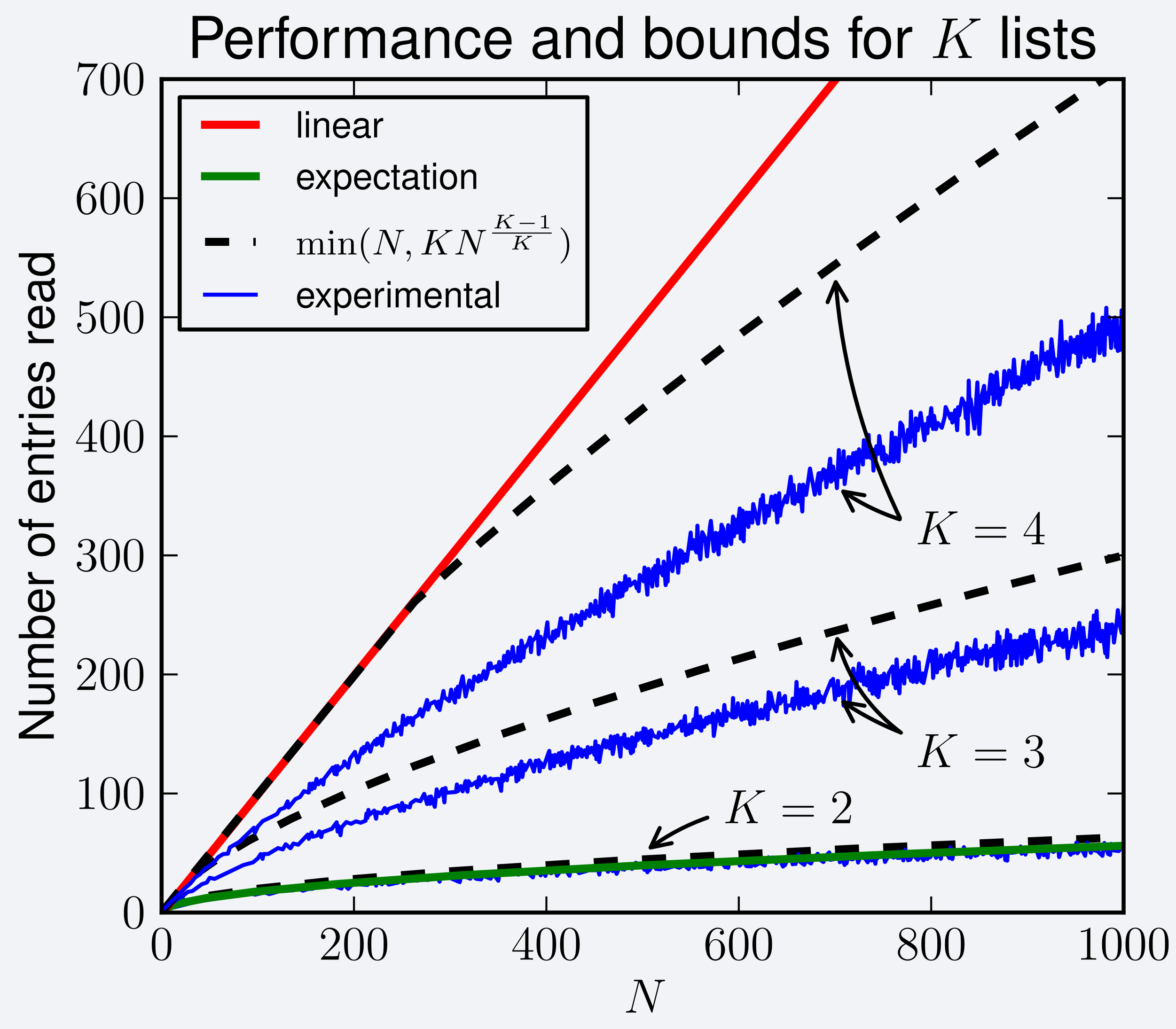
Our results

The consequences of this result are as follows:

- We are able to lower the asymptotic expected running time of the Junction-Tree Algorithm for *any* graphical model whose cliques factorize into lower-order terms.
- For any cliques composed of pairwise factors, we obtain an expected speed-up of *at least* $\Omega(\sqrt{N})$ (assuming N states per node).
- For cliques composed of K-ary factors, the expected speed-up becomes at least $\Omega(\frac{1}{K}N^{\frac{1}{K}})$, though it is *never asymptotically slower* than the original solution.
- The expected-case improvement is achieved when the conditional densities of different factors are uncorrelated.
- If the conditional densities are positively correlated, the performance will be better than the expected case.
- If the conditional densities are negatively correlated, the performance will be worse than the expected case, but is never asymptotically more expensive than the traditional Junction-Tree Algorithm.

Full details of our method can be found on [4].

Results

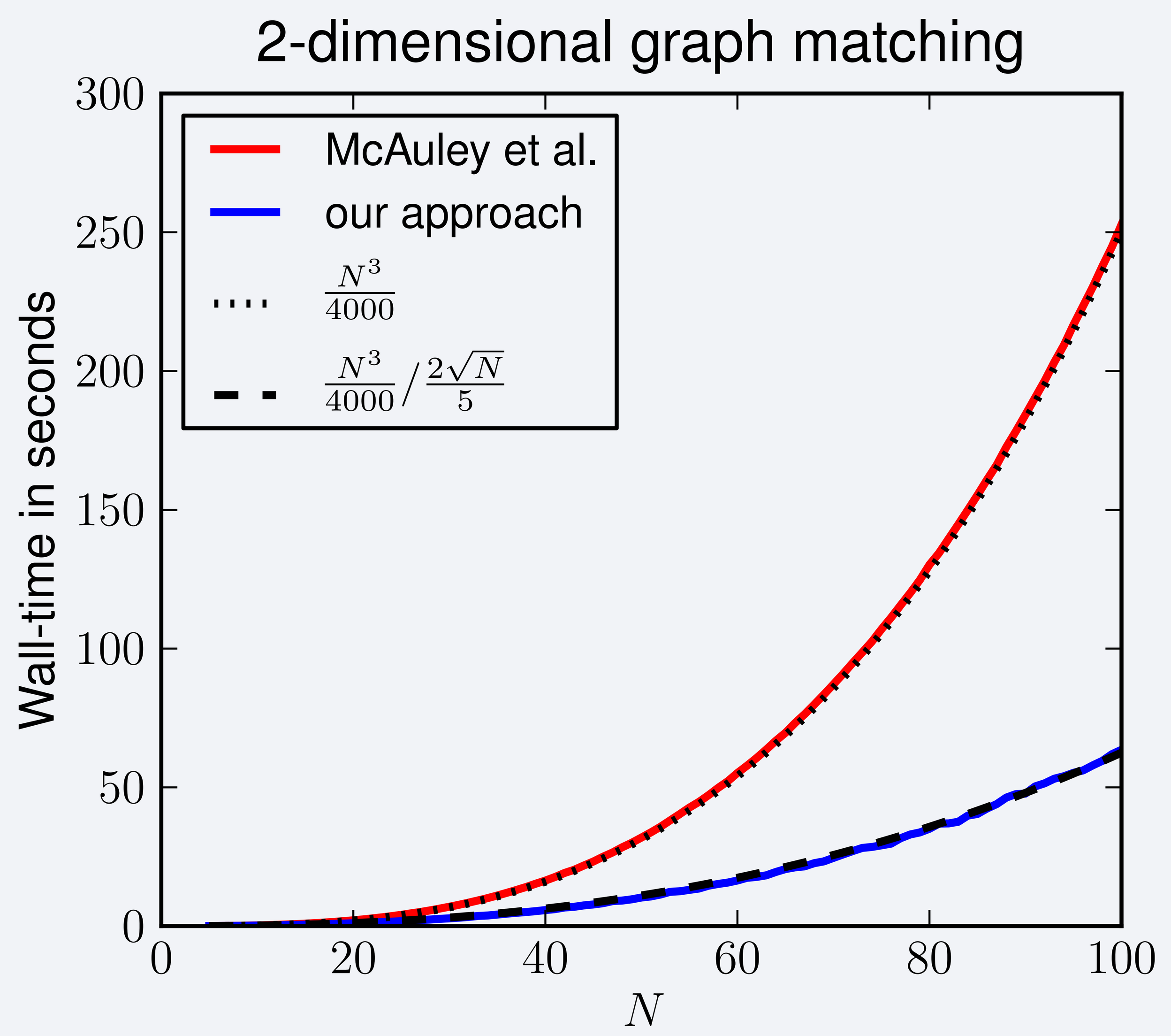


The above plot shows the savings our method obtains (compared to the linear-time solution) when used to solve

$$\hat{i} = \underset{i \in \{1 \ldots N\}}{\operatorname{argmax}} \{\mathbf{v}_1[i] \times \mathbf{v}_2[i] \times \cdots \times \mathbf{v}_K[i]\}.$$

This saving is obtained whenever we have K^{th}-order factors in cliques with more than K terms.

Graph matching



It is possible to specify a model for *graph matching* that includes second-order factors within third-order cliques [5]. If we are searching for a graph of size M withing a graph of size N, the algorithm of [5] has a running time of $O(MN^3)$; our results improve this to $O(MN^2\sqrt{N})$. The above plot shows the actual running time of both methods, which demonstrates that our approach has minimal computational overhead, and is beneficial even for very small values of N.

Bibliography

[1] Leonid Sigal and Michael J. Black.
Predicting 3d people from 2d pictures.
In *AMDO*, 2006.

[2] Michel Galley.
A skip-chain conditional random field for ranking meeting utterances by importance.
In *EMNLP*, 2006.

[3] James M. Coughlan and Sabino J. Ferreira.
Finding deformable shapes using loopy belief propagation.
In *ECCV*, 2002.

[4] Julian J. McAuley and Tibério S. Caetano.
Exact inference in graphical models: is there more to it?
Technical report, arXiv preprint: cs.AI/0910.3301, 2009.

[5] J. J. McAuley, T. S. Caetano, and M. S. Barbosa.
Graph rigidity, cyclic belief propagation and point pattern matching.
IEEE Trans. on PAMI, 30(11):2047–2054, 2008.

julian.mcauley@nicta.com.au, tiberio.caetano@nicta.com.au