Exploiting data-independence for fast belief-propagation

Julian McAuley, Tibério Caetano

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

Abstract

MAP-inference in graphical models requires that we maximize the sum of two terms: a *data-dependent* term, encoding the conditional likelihood of a certain labeling given an observation, and a *data-independent* term, encoding some prior on labelings. **Often, the data-dependent factors contain fewer latent variables than the data-independent factors. We note that MAP-inference in any such graphical model can be made substantially faster by appropriately pre-processing its data-independent terms.** Our main result is to show that message-passing in any such pairwise model has an expected-case exponent of only 1.5 on the number of states per node, leading to significantly faster algorithms than the standard quadratic time solution.

'Data-Independence'

MAP-inference in a graphical model $\mathcal{G}$ consists of solving an optimization problem of the form

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \sum_{C \in \mathcal{C}} \Phi_C(\mathbf{y}_C),$$

where $\mathcal{C}$ is the set of cliques in the model. Often, the model can be further factorized if we make a distinction between the latent variables $\mathbf{y}$ and the observation $\mathbf{x}$:

$$\hat{\mathbf{y}}(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmax}} \underbrace{\sum_{F \in \mathcal{F}} \Phi_F(\mathbf{y}_F|\mathbf{x}_F)}_{\text{data dependent}} + \underbrace{\sum_{C \in \mathcal{C}} \Phi_C(\mathbf{y}_C)}_{\text{data independent}} .$$

We say that those cliques containing only latent variables are **data-independent**. In many models, **those cliques that contain an observed variable contain fewer latent variables than the purely latent cliques**, i.e., each $F \in \mathcal{F}$ is a proper subset of some $C \in \mathcal{C}$. Examples of such models are shown at top-right.

Bibliography

[1] Julian J. McAuley and Tibério S. Caetano. Exploiting within-clique factorizations in junction-tree algorithms. *AISTATS*, 2010.

[2] U. Quasthoff, M. Richter, and C. Biemann. Corpus portal for search in monolingual corpora. In *Language Resources and Evaluation*, 2006.

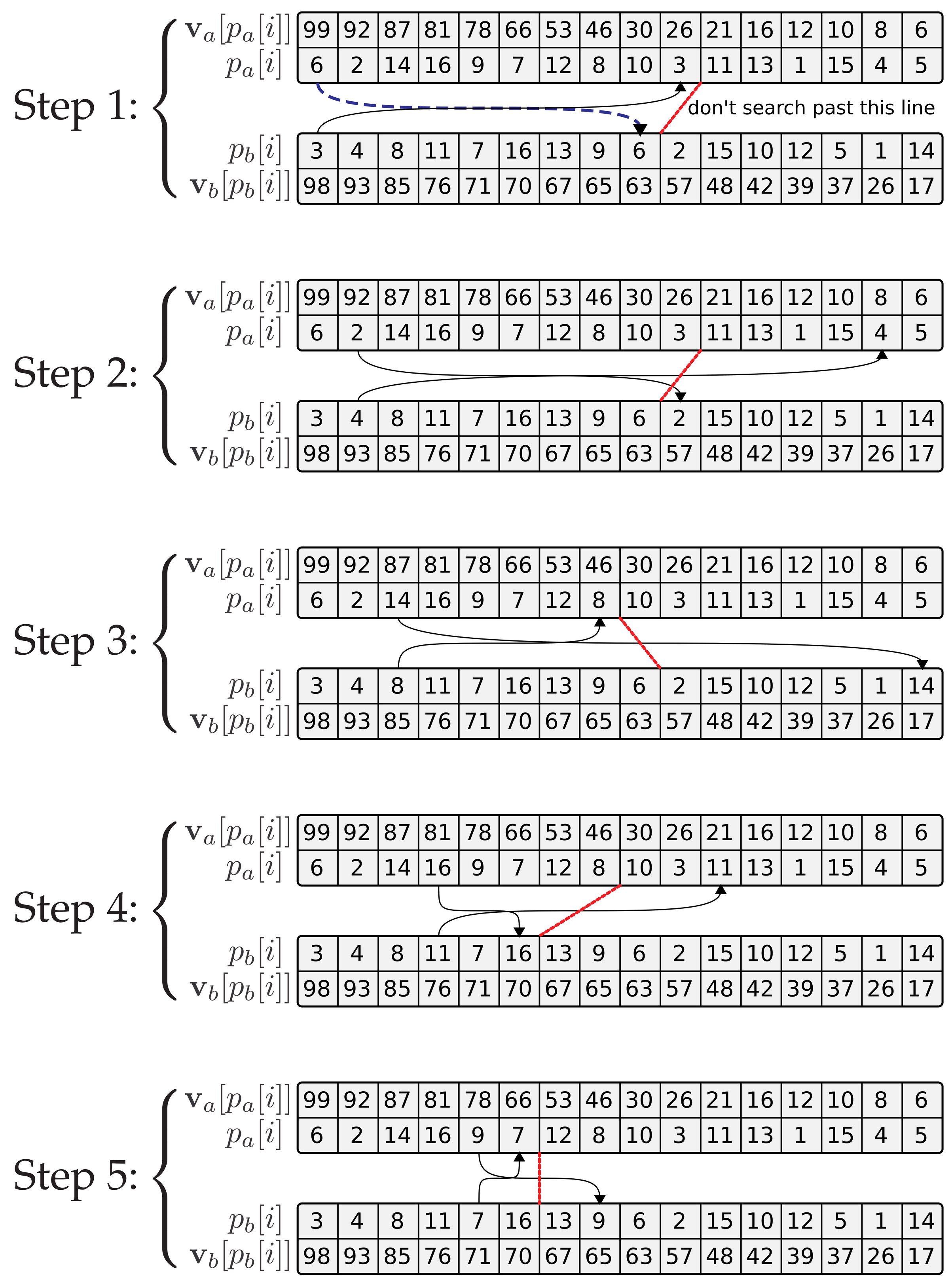
Message-Passing

In these models, message passing between two cliques $A = (i, j)$, $B = (j, k)$ takes the form

$$m_{A \to B}(y_i) = \Psi_i(y_i) + \max_{y_j} \Psi_j(y_j) + \Phi_{i,j}(y_i, y_j), \quad (1)$$

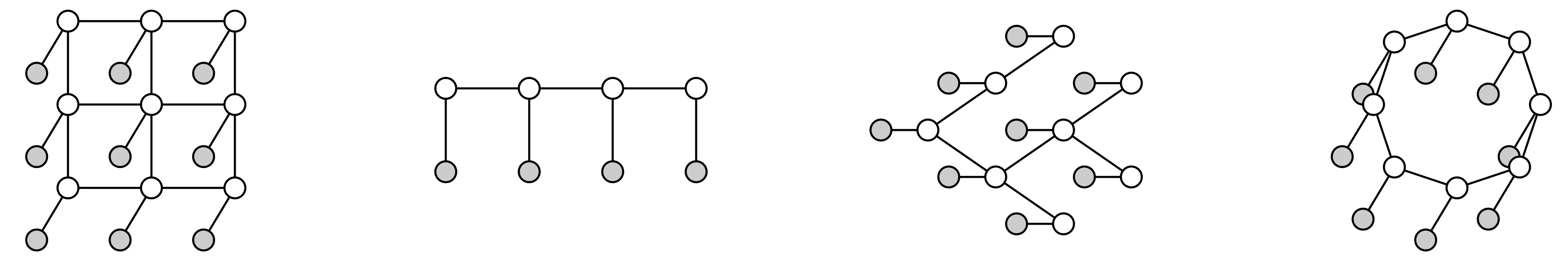
which is equivalent to **matrix-vector multiplication in the max-sum semiring**. In a recent paper [1], we showed that matrix-matrix multiplication in this semiring takes $O(N^{2.5})$ (for $N \times N$ matrices). In our current work, we note that a similar result can be applied to matrix-vector multiplication, **so long as the matrix is known in advance**. Since the 'matrix' in the above equation simply encodes a prior, it can be preprocessed *offline*.

How it Works



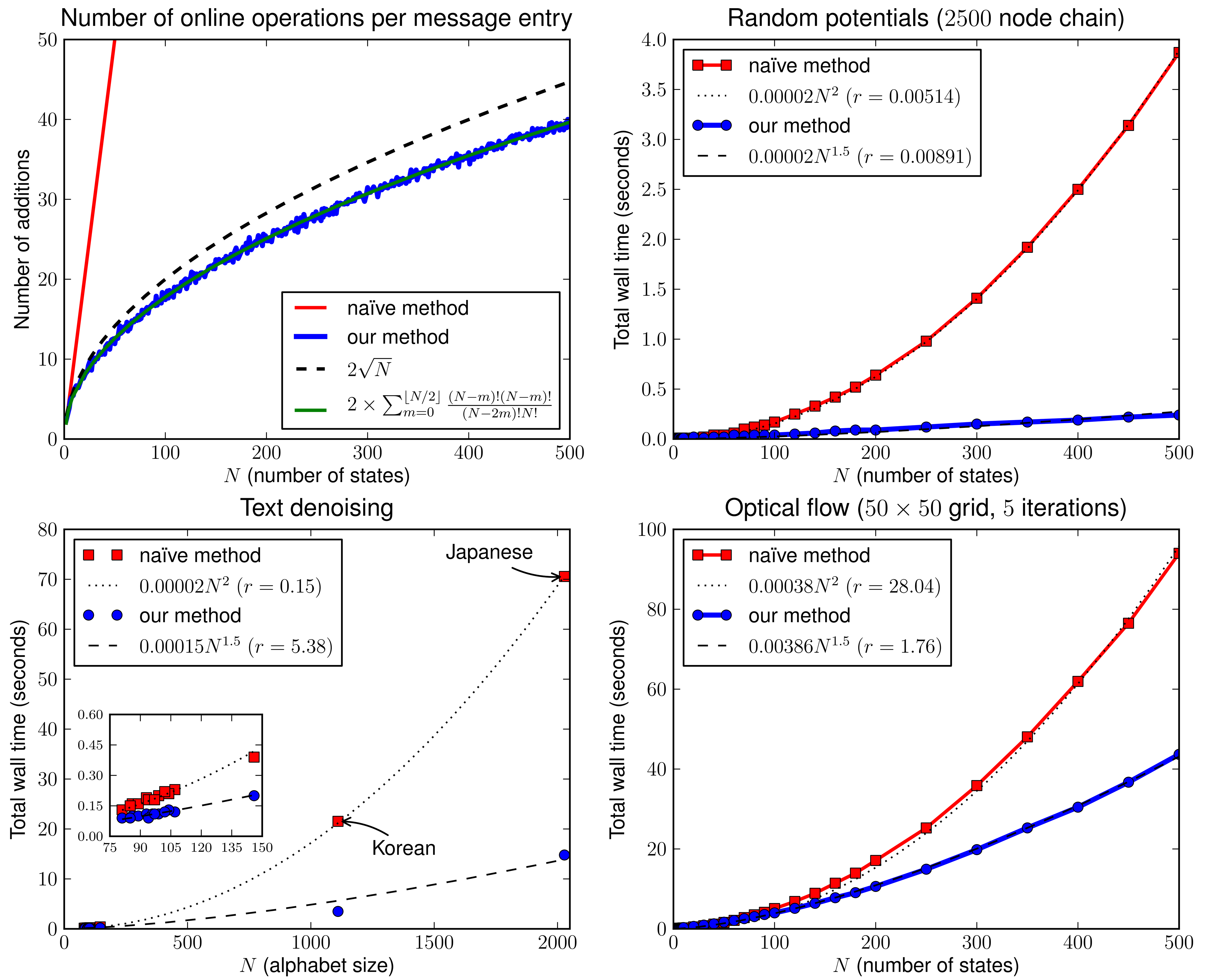
We wish to compute $\max_i \mathbf{v}_a[i] + \mathbf{v}_b[i]$. Arrows connect corresponding elements of $\mathbf{v}_a$ and $\mathbf{v}_b$, as sorted by p_a and p_b. We draw a red line connecting the leftmost arrowheads that have been seen so far. Any 'arrows' whose tail lies to the right of this line cannot possibly correspond to an optimal solution.

Example Models



Examples of graphical models to which our results apply: **cliques containing observations have fewer latent variables than purely latent cliques**. In other words, cliques containing a grey node encode the *data likelihood*, whereas cliques containing only white nodes encode *priors*. We focus on cases where the gray nodes have degree one (i.e., they are connected to only one white node). In such cases we obtain an $\Omega(\sqrt{N})$ speedup on the number of states per node.

Experiments



Top left: number of operations required to compute each entry of the message vector (eq. 1). Top right: message passing in a chain-structured model with uniform potentials. Bottom left: typo-correction in a chain-structured model [2]. Bottom right: optical flow in a grid.