HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

# AVERAGING OF DECOMPOSABLE GRAPHS BY DYNAMIC PROGRAMMING AND SAMPLING

Kustaa Kangas, Teppo Niinimäki, Mikko Koivisto
Department of Computer Science

We present algorithms for Bayesian learning of decomposable models from data. Priors of a certain form admit exact averaging in $O(3^n n^3)$ time and sampling $T$ graphs from the posterior in $O(4^n + nT)$ time. To target a broader class of priors we associate each sample with an importance weight. Empirically, we compare averaging to optimization and demonstrate the accuracy of our importance sampling estimates.

## PROBLEM

### DECOMPOSABLE MODEL

- Undirected graph $\mathcal{G}$ on $V = \{1, \ldots, n\}$ with a *junction tree*.
- A joint distribution $p(x_1, \ldots, x_n)$ that factorizes as

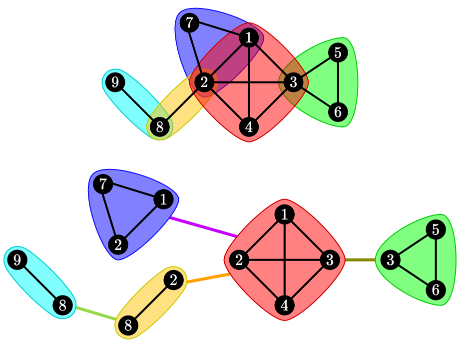
$$p(x_V) = \frac{\prod_{C \in \mathcal{C}} p(x_C)}{\prod_{S \in \mathcal{S}} p(x_S)},$$

where

$\mathcal{C}$ is the set of *cliques* of $\mathcal{G}$,

$\mathcal{S}$ is the multiset of *separators*,

$x_C = (x_v : v \in C)$.



$$\frac{\varphi_c(2,8)\varphi_c(1,2,7)\varphi_c(8,9)\varphi_c(1,2,3,4)\varphi_c(3,5,6)}{\varphi_s(8)\varphi_s(1,2)\varphi_s(2)\varphi_s(3)}$$

### BAYESIAN LEARNING

**Problem:** Compute the *marginal*

$$Z_\varphi = \sum_{\mathcal{G}} \varphi(\mathcal{G}),$$

where $\varphi$ is a *decomposable function*, i.e.,

$$\varphi(\mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} \varphi_c(C)}{\prod_{S \in \mathcal{S}} \varphi_s(S)}, \quad \varphi_c, \varphi_s : 2^V \to \mathbb{R}.$$

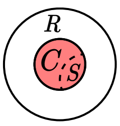
**Example (feature posterior probability):** The posterior probability of a feature $\psi(\mathcal{G})$ is the expectation $\mathrm{E}(\psi(\mathcal{G}) \mid \text{data}) = Z_{\pi\psi}$ where $\pi(\mathcal{G}) \propto \Pr(\text{data} \mid \mathcal{G}) \Pr(\mathcal{G})$.

## EXACT AVERAGING

Computing $Z_\varphi$ for a decomposable $\varphi$ is doable up to $n \simeq 8$. It is more efficient to compute

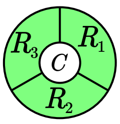
$$Z_{\varphi\tau} = \sum_{\mathcal{G}} \varphi(\mathcal{G}) \tau(\mathcal{G}),$$

where $\tau(\mathcal{G})$ denotes the number of rooted junction trees of $\mathcal{G}$. We get $Z_{\varphi\tau} = f(\emptyset, V)$, where $f$ is obtained by adapting a recent dynamic programming algorithm [1]:



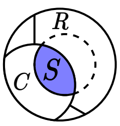

1. Given sets $S$ and $R$, choose a clique $C$ within $R$ that contains $S$ completely:

$$f(S,R) = \sum_{S \subset C \subseteq S \cup R} \varphi_c(C)\, g(C, R \setminus C)$$




2. Partition the remaining vertices into $R_1, \ldots, R_k$:

$$g(C,U) = \sum_{\min U \in R \subseteq U} h(C,R)\, g(C, U \setminus R)$$




3. For each part $R$ choose a separator $S$ contained in $C$. Recurse back to step 1:

$$h(C,R) = \sum_{S \subset C} f(S,R) / \varphi_s(S)$$

The algorithm runs in $O(4^n)$ time and $O(3^n)$ space. By using variants of fast zeta transform and fast subset convolution, we can bring the running time to $O(3^n n^3)$.

## MONTE CARLO AVERAGING

To estimate the posterior expectation $Z_{\pi\psi}$:

1. Run the exact algorithm for a $\varphi \propto \pi$.
2. Backtrack into the recurrences to draw samples $\mathcal{G}_1, \ldots, \mathcal{G}_T$ from $\pi' \propto \pi\tau$.
   (a) *Fixed bucketing*: tunable time/space tradeoff between $O(2^n)/O(3^n)$ and $O(n)/O(4^n)$; or
   (b) *Adaptive caching*: good if $\pi$ is skewed.
3. Obtain the self-normalized importance sampling estimate

$$\tilde{Z}_{\pi\psi} = \frac{\sum_{i=1}^T w_i \psi(\mathcal{G}_i)}{\sum_{i=1}^T w_i}, \quad w_i = \frac{1}{\tau(\mathcal{G}_i)}.$$

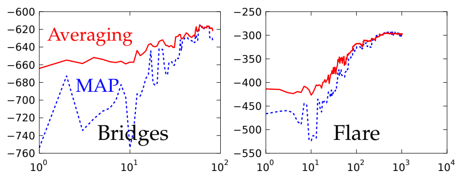
Computing $\tau(\mathcal{G}_i)$ takes $O(n^2)$ time [2].

### EXPERIMENTS

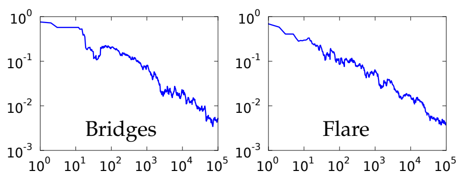
We ran our implementation [3] on datasets of a varying number of variables ($n$) and records ($m$). The table shows the seconds spent in the exact algorithm (E) and sampling $10^7$ graphs (S).

| Dataset | $n$ | $m$ | E | S |
|---|---|---|---|---|
| Asia | 8 | 10000 | 0.018 | 33 |
| Bridges | 12 | 108 | 5.6 | 54 |
| Flare | 13 | 1066 | 24 | 77 |
| Votes | 17 | 435 | 7497 | 167 |

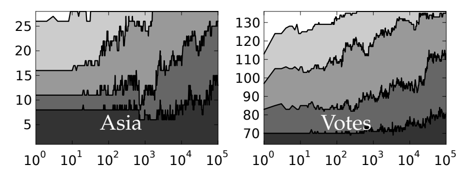
**Prediction accuracy – averaging vs. MAP:** The log probability (y-axis) of a test set for a varying amount of training data (x-axis).




**Edge posterior probabilities:** The largest estimation error over all edges (y-axis) for a varying number of samples (x-axis).




The number of edges (y-axis) with an error at most $10^{-1}$, $10^{-2}$, $10^{-3}$, and $10^{-4}$, for a varying number of samples (x-axis).




[1] K. Kangas, M. Koivisto, and T. Niinimäki. *Learning chordal Markov networks by dynamic programming*. NIPS, 2014.
[2] A. Thomas and P. J. Green. *Enumerating the junction trees of a decomposable graph*. J. Comp. Graph. Stat., 2009.
[3] *Junctor/Adjunct*. Available at www.cs.helsinki.fi/u/jwkangas/junctor.

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY