Exploiting the graphical model structure of latent Gaussian models

Helen Ogden

i-like.org.uk

The likelihood for a latent Gaussian model is a high-dimensional integral. We exploit the structure of the integrand to reduce the cost of computing an accurate approximation of the likelihood, by modifying methods designed for graphical models with discrete variables.

Example: pairwise competition models

We observe binary matches Y_{ij} played between pairs of players, and model

$$Pr(Y_{ij} = 1|\lambda_i, \lambda_j) = \text{logit}^{-1}(\lambda_i - \lambda_j),$$

where λ_i is a latent 'ability' of player i. We have covariates x_i for player i, and model $\lambda_i = \beta x_i + \sigma u_i$, where $u_i \sim N(0,1)$. The likelihood is

$$L(\theta; y) = \int_{\mathbb{R}^n} \prod_{i \sim j} \text{logit}^{-1}\left\{y_{ij}\left[\beta(x_i - x_j) + \sigma(u_i - u_j)\right]\right\} \prod_{i=1}^{n} \phi(u_i) d\mathbf{u},$$

where $i \sim j$ indicates a pair of players who compete against each other.

This model is an example of a **latent Gaussian model**. We develop methods to **approximate the likelihood**, which may be used across this wider class of models.

Laplace approximation to the likelihood

The **Laplace approximation** to the likelihood replaces the integrand

$$g(\mathbf{u}; \theta) = \prod_{i \sim j} \text{logit}^{-1}\left\{y_{ij}\left[\beta(x_i - x_j) + \sigma(u_i - u_j)\right]\right\} \prod_{i=1}^{n} \phi(u_i)$$

with a normal approximation: $\tilde{g}(\mathbf{u}; \theta) = c_\theta\, \phi(\mathbf{u}, \mu_\theta, \Sigma_\theta)$, giving $\tilde{L}(\theta; y) \approx c_\theta$. In our example, the Laplace approximation should be good if each player competes in a lot of matches, but may be **poor if each player competes in a small number of matches**.

Graphical models

Graphical models are distributions for $\mathbf{u} = (u_1, \dots u_n)$ such that

$$p(\mathbf{u}) \propto g(\mathbf{u}) = \prod_C g_C(\mathbf{u}_C).$$

We represent this **factorization structure** on a **dependence graph**, with one node per variable, and an edge between any two variables contained within the same term of the factorization. Variables not joined by an edge are **conditionally independent**, given all the other variables.

Rephrasing the problem

The likelihood for the pairwise competition model is the **normalizing constant** Z corresponding to a graphical model. The dependence graph has a node for each player, and an edge for each match. The Laplace approximation may fail if this dependence graph is **sparse**, so we focus on this case.

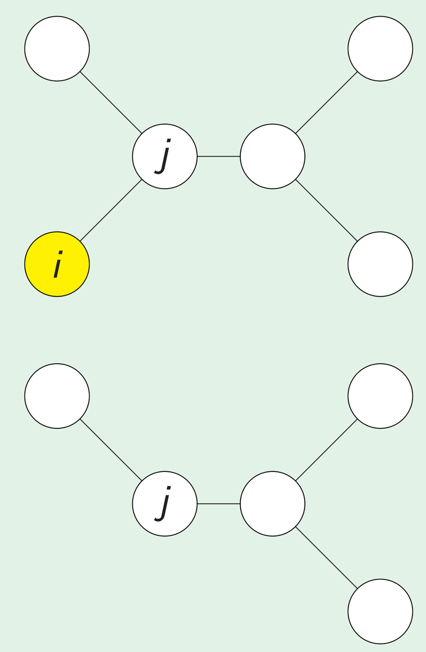
Variable elimination with discrete variables

Suppose we have a graphical model where each $u_i \in \{0,1\}$, and we want to compute $Z = \sum_{\mathbf{u}} g(\mathbf{u})$. Naively, we could just sum 2^n contributions. We can reduce that cost by exploiting the structure of the dependence graph.

If the dependence graph is a **tree** (has no cycles), we can compute Z at cost **linear** in n.

1. Choose a 'leaf' variable i.
2. Sum over u_i: it is involved in only one term $g_{ij}(u_i, u_j)$ of the factorization. Compute $\bar{g}_j(u_j) = \sum_{u_i} g_{ij}(u_i, u_j)$, and replace g_{ij} with $\bar{g}_j$ in the factorization.
3. The resulting marginal distribution for the remaining variables is a graphical model with new tree-structured dependence graph.

Repeat until all variables removed.



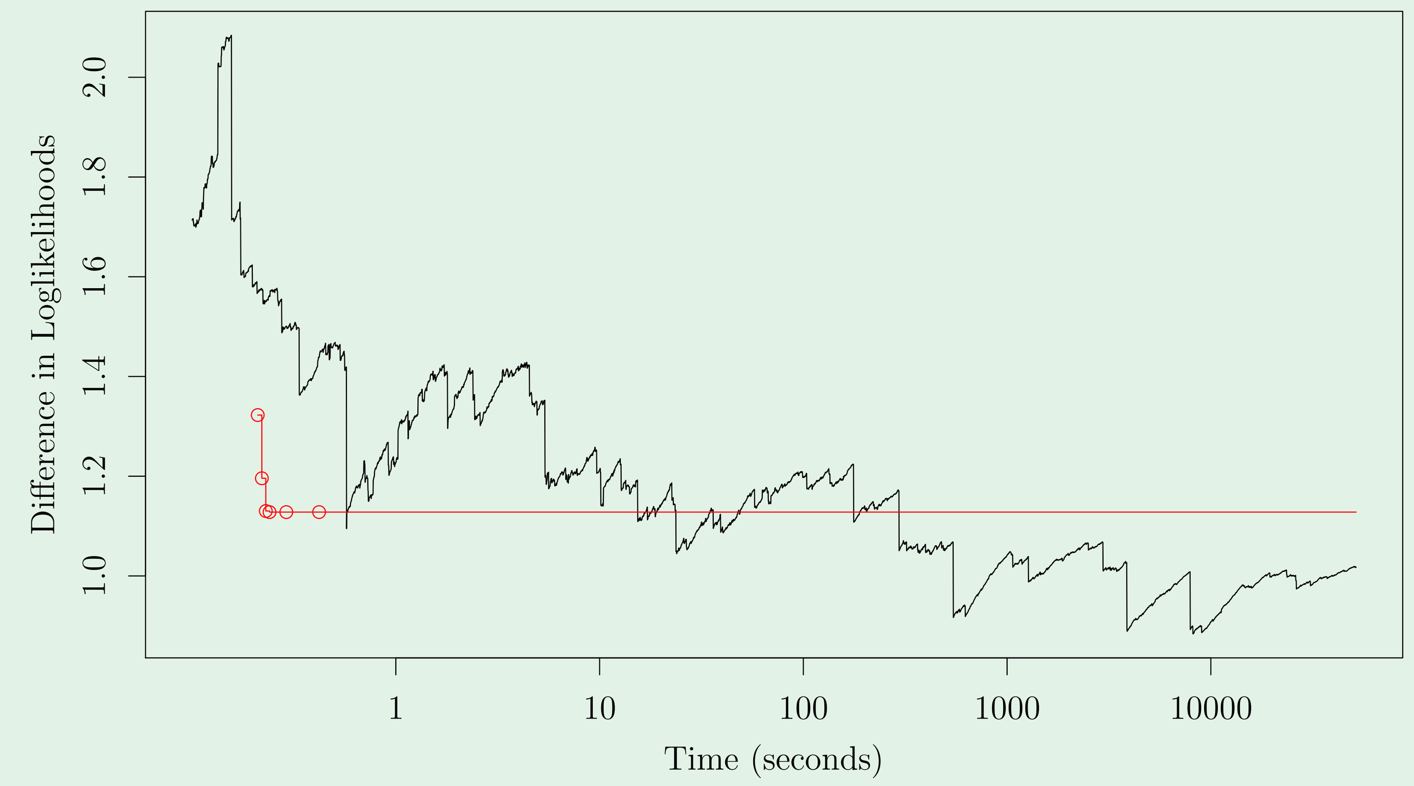
This is a special case of the **variable elimination** algorithm. More generally, a step of the algorithm involves computing $\bar{g}_{N_i}(\mathbf{u}_{N_i}) = \sum_{u_i} \prod_{C: i \in C} g_C(\mathbf{u}_C)$, at cost $O(2^{|N_i|})$, where N_i are the neighbours of i in the dependence graph. We get a factorization over a new dependence graph, with i removed and N_i joined together. See [2] for details.

Variable elimination with continuous variables

At each stage, we must compute $\bar{g}_{N_i}(\mathbf{u}_{N_i}) = \int_{-\infty}^{\infty} \prod_{C: i \in C} g_C(\mathbf{u}_C) du_i$. For each fixed u_{N_i}, we can use numerical integration to approximate $\bar{g}_{N_i}(u_{N_i})$. We want an **approximate representation of the function** $\bar{g}_{N_i}(.)$, so we store $\bar{g}_{N_i}(u_{N_i})$ at a fixed set of points u_{N_i}, and **interpolate** between those points. The approximate representation is based on the normal approximation used to construct the Laplace approximation, so the method is always at least as accurate as the Laplace approximation. See [3] for details.

Comparing likelihood approximations

For a tree-structured pairwise competition model with $n = 63$ players, we approximate the difference in the loglikelihood at two points close to the MLE.



The black line is an importance sampling approximation, based on the same normal distribution used to construct the Laplace approximation. The red line is the approximation based on variable elimination, as the number of points used for storage increases.

Sparse but non-tree-like graphs

Many latent Gaussian models have dependence graphs which are 'close' to trees, and variable elimination works well.

However, challenging cases remain: if we have a graphical model with binary variables, and a square **lattice** dependence graph, the cost of computing Z with variable elimination is $O(n2^{\sqrt{n}})$.

I have been working on a new approximation to Z in these cases, controlled by a parameter t. For a square lattice, the cost is $O(n2^t)$, and the error shrinks quickly with t. I hope to combine this new approximation with approximate function storage to get better approximations to the likelihood in latent Gaussian models. See [1].

References

[1] https://github.com/heogden/rgraphpass.

[2] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* The MIT Press, 2009.

[3] Helen E. Ogden. A sequential reduction method for inference in generalized linear mixed models. *Electron. J. Statist.*, 9:135–152, 2015.

h.ogden@warwick.ac.uk