Discrete Markov Random Fields

Possibilities and computational challenges

Nial Friel

University College Dublin
`nial.friel@ucd.ie`

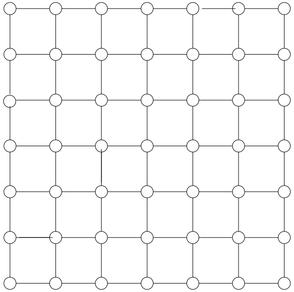
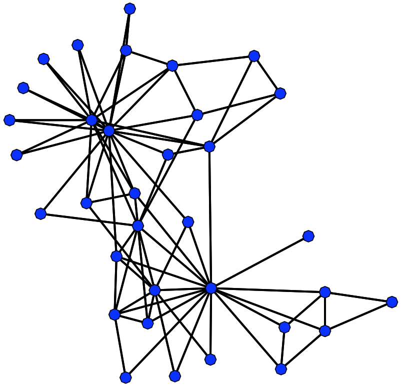
August, 2009

Just to set the record straight...

THE CELTIC FOOTBALL CLUB
1888

ΠΑΝΑΘΗΝΑΪΚΟΣ
1908

Generally, discrete Markov random fields are distributions defined on graphs:

The graphs may be regular or not.

1. Continuous-valued Markov random fields, eg Gaussian MRFs
2. Discrete-valued Markov random fields
   - Regular lattices – image models; spatial statistics; (Lecture 1)
   - Irregular lattices – social network models; classification; (Lecture 2)

Part I – MRFs on regular lattices.

Binary MRFs on regular lattices

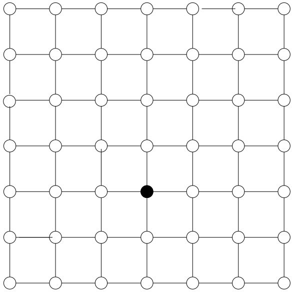
- Defined on a lattice $\mathbf{x} = \{x_1, \ldots, x_n\}$.
- Lattice points x_i take values $\{-1, 1\}$.
- Full conditional $p(x_i | x_{-i}, \theta) = p(x_i | \text{neighbours of } i, \theta)$.

$$p(x|\theta) \propto q(x|\theta) = \exp\left\{ \theta_0 \sum_i x_i + \frac{1}{2}\theta_1 \sum_{i \sim j} x_i x_j \right\}.$$

Here $\sim$ means "is a neighbour of".

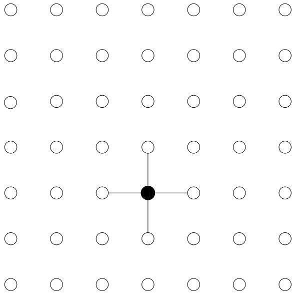
Binary MRFs on regular lattices

The Markov property



Binary MRFs on regular lattices

The Markov property



Binary MRFs on regular lattices

The normalising constant is typically *difficult* to compute:

$$z(\theta) = \sum_{x_1} \cdots \sum_{x_n} q(x|\theta).$$

Our problem of interest

How can statistical inference be carried for a model

$$p(\mathbf{x}|\theta) = \frac{q(\mathbf{x}|\theta)}{z(\theta)},$$

where $z(\theta)$ is an intractable normalising constant?

Maximum likelihood estimation:

$$\hat{\theta} = \arg\max_{\theta} p(\mathbf{x}|\theta) = \arg\max_{\theta} \frac{q(\mathbf{x}|\theta)}{z(\theta)}.$$

Bayesian inference:

Here we use the posterior distribution $p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta)$.
A Metropolis-Hastings MCMC scheme requires calculation of

$$\frac{p(\mathbf{x}|\theta^*)p(\theta^*)}{p(\mathbf{x}|\theta)p(\theta)} = \frac{q(\mathbf{x}|\theta^*)p(\theta)}{q(\mathbf{x}|\theta)p(\theta)} \frac{z(\theta)}{z(\theta^*)}.$$

Posterior distributions of the type

$$\begin{aligned} p(\theta|\mathbf{x}) &\propto p(\mathbf{x}|\theta)p(\theta) \\ &= \frac{q(\mathbf{x}|\theta)}{z(\theta)}p(\theta) \end{aligned}$$

are sometimes called doubly intractable distributions.

Posterior distributions of the type

$$\begin{aligned} p(\theta|\mathbf{x}) &\propto p(\mathbf{x}|\theta)p(\theta) \\ &= \frac{q(\mathbf{x}|\theta)}{z(\theta)}p(\theta) \end{aligned}$$

are sometimes called doubly intractable distributions.

This type of complication occurs frequently for *Markov random field* models.

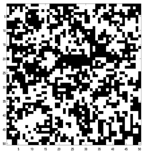
An historical aside

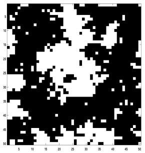
- The Metropolis algorithm (1953) arose from the need to sample from $p(\mathbf{x}|\theta)$.
- Geman and Geman (1984) illustrated the Gibbs sampler for MRFs. The Gibbs sampler was later popularised by Gelfand and Smith (1990).
- Perfect sampling, Coupling from the past: Propp and Wilson (1996) showed that it's possible to use MCMC to sample *exactly* from an MRF.

All of these seminal papers perform MCMC sampling for the MRF x conditional on θ.

Realisations of binary MRfs

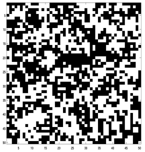
As the parameter θ increases, the level of spatial aggregation does too.

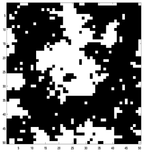




Realisations of binary MRfs

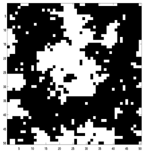
As the parameter θ increases, the level of spatial aggregation does too.

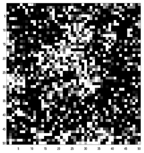




Hidden MRFs

Here a true scene $\mathbf{x}$ is corrupted by a noise process with parameters μ yielding data $\mathbf{y}$. The aim to infer all unknown parameters.





Standard MCMC algorithm

Step 1. Update each x_i in turn by Gibbs sampling from:

$$p(x_i|\mathbf{x}_{\setminus i}, \mathbf{y}, \theta, \mu) \propto p(y_i|x_i, \mu)p(x_i|x_{N(i)}, \theta). \qquad (1)$$

Standard MCMC algorithm

STEP 1. UPDATE EACH x_i IN TURN BY GIBBS SAMPLING FROM:

$$p(x_i|\mathbf{x}_{\backslash i}, \mathbf{y}, \theta, \mu) \propto p(y_i|x_i, \mu)p(x_i|x_{N(i)}, \theta). \qquad (1)$$

STEP 2. UPDATE μ: CARRY OUT A M-H UPDATE OF μ FROM THE FULL CONDITIONAL:

$$p(\mu|\mathbf{x}, \mathbf{y}, \theta) \propto \left\{ \prod_{i=1}^{n} p(y_i|x_i, \mu) \right\} \pi_\mu(\mu).$$

Standard MCMC algorithm

Step 1. Update each x_i in turn by Gibbs sampling from:

$$p(x_i|\mathbf{x}_{\setminus i},\mathbf{y},\theta,\mu) \propto p(y_i|x_i,\mu)p(x_i|x_{N(i)},\theta). \qquad (1)$$

Step 2. Update μ: Carry out a M-H update of μ from the full conditional:

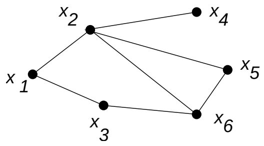
$$p(\mu|\mathbf{x},\mathbf{y},\theta) \propto \left\{\prod_{i=1}^{n} p(y_i|x_i,\mu)\right\}\pi_\mu(\mu).$$

Step 3. Update θ: Carry out a M-H update of θ from the full conditional:

$$p(\theta|\mathbf{x},\mu,\mathbf{y}) \propto p(\mathbf{x}|\theta)\pi_\theta(\theta).$$

We now concentrate on how to deal with the intractable normalising constant $z(\theta)$.

Undirected Graphs and joint distributions



The joint distribution can be written as

$$p(x_1, \ldots, x_6) = \frac{1}{z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_6)\psi(x_2, x_5, x_6).$$

Naively, the normalising constant is computed as

$$z = \sum_{x_1} \cdots \sum_{x_6} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_3, x_6)\psi(x_2, x_5, x_6).$$

Computational complexity scales as s^6 (assuming x_i has s states).

Undirected Graphs and joint distributions

However,

$$z = \sum_{x_1}\sum_{x_2}\psi(x_1,x_2)\sum_{x_3}\psi(x_1,x_3)\sum_{x_4}\psi(x_2,x_4) \sum_{x_6}\psi(x_3,x_6)\sum_{x_5}\psi(x_2,x_5,x_6).$$

No more than 3 terms appear in any summand. Computational complexity is decreased!

In general we would like to perform the summation so that the largest factor is as small as possible.

MRF in factorisable form

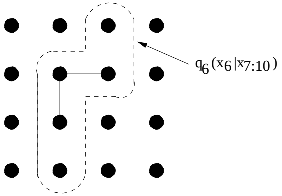
Define an index $i = 1, \ldots, n$, where points are ordered from top to bottom and rows from left to right. m denotes the number of rows.

$$q(x|\theta) = q_n(x_n|\theta) \prod_{i=1}^{n-1} q_i(x_i|x_{i+1:i+m}, \theta),$$

where we define

$$q_i(x_i|x_{i+1:i+m}, \theta) = \exp(\theta_0 x_i + \theta_1 x_i (x_{i+1} + x_{m+i}))$$

with modifications when i corresponds to a point on the last row or column.



MRF in factorisable form

We use the shorthand notation $x_{i:j} = (x_i, \dots, x_j)$.

$$\begin{aligned} z(\theta) &= \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} q_1(x_1 | x_{2:m+1}, \theta) q_2(x_2 | x_{3:m+2}, \theta) \dots q_n(x_n | \theta) \\ &= \sum_{x_1} q_1(x_1 | x_{2:m+1}, \theta) \sum_{x_2} q_2(x_2 | x_{3:m+2}, \theta) \cdots \sum_{x_n} q_n(x_n | \theta). \end{aligned}$$

The recursive algorithm

$$\begin{aligned}
z_1(\theta, x_{2:n}) &= \sum_{x_1} q_1(x_{1:m+1}, \theta) \\
z_i(\theta, x_{i+1:n}) &= \sum_{x_i} q_i(x_{i:m+i}, \theta) z_{i-1}(\theta, x_{i:n}), \text{ for } i = 2, \ldots, n. \\
z(\theta) &= z_n(\theta)
\end{aligned}$$

Exact sampling: The recursive algorithm

$$\begin{aligned} z_1(\theta, x_{2:n}) &= \sum_{x_1} q_1(x_{1:m+1}, \theta) \\ z_i(\theta, x_{i+1:n}) &= \sum_{x_i} q_i(x_{i:m+i}, \theta) z_{i-1}(\theta, x_{i:n}), \text{ for } i = 2, \ldots, n. \end{aligned}$$

Effectively z_i is the normalising constant for

$$p(x_{1:i}|x_{i+1:n}, \theta) \propto q(x_{1:i}|x_{i+1:n}, \theta)$$

Each z_i depends on m variables $x_{i+1:m+1}$ - In total there are 2^m!

Exact sampling: The recursive algorithm

$$p(x|\theta) = p(x_1|x_{2:n},\theta)p(x_2|x_{3:n},\theta)\dots p(x_n|\theta).$$

We gather a sample from $p(x|\theta)$ by sampling from

$$\begin{aligned} p(x_i|x_{i+1:n},\theta) &= \frac{p(x_{1:i}|x_{i+1:n},\theta)}{p(x_{1:i-1}|x_{i:n},\theta)} \\ &= \frac{q(x_{1:i}|x_{i+1:n},\theta)z_{i-1}(\theta,x_{i:n})}{q(x_{1:i-1}|x_{i:n},\theta)z_i(\theta,x_{i+1:n})}, \end{aligned}$$

for $i = n, n-1, \dots, 1$.

Exact sampling: The recursive algorithm

We propose a two pass algorithm:

Forwards pass: Using the recursive scheme above, we generate in turn each $z_i(\theta, x_{i+1:n})$ for $i = 1, 2, \ldots, n$.

Backwards pass: Sample x_i from $p(x_i | x_{i+1:n}, \theta)$ using the z_i's, for $i = n, n-1, \ldots, 1$.

Computer implementation

- Main computational loads arises from generating the collection of z_i's from the forwards pass.
  For each z_i, there are 2^m realisations, in total $n \times 2^m$.
- In our computer implementation, we can sample lattices where the smaller dimension is ≤ 19. For example, a 19×19 lattice takes about 150 seconds.

Extensions of the algorithm

The algorithm can be extended to:

1. sample from $p(x|\theta)$.
2. sample from $p(x|\theta, y)$, where y is a hidden version of x.
3. compute the modal lattice for $p(x|\theta, y)$. Again a two pass algorithm is used, essentially sampling from an annealed distribution at temperature 0.
4. compute the marginal distribution of points, pairs of points, eg $p(x_i|\theta)$, $p(x_i, x_j|\theta, y)$ for neighbours i, j.
5. different neighbourhood structures.
6. more than 2 states.

Further extensions: hidden MRFs

Consider the posterior marginal for θ. For any realisation x,

$$p(\theta|y) = \frac{p(x, \theta|y)}{p(x|\theta, y)}.$$

We can write this as

$$p(\theta|y) \propto \frac{p(y|x)p(x|\theta)p(\theta)}{p(x|\theta, y)},$$

the normalising constant is $p(y)$ - the marginal likelihood.

Each term on the RHS above can be calculated exactly.

We estimate $p(\theta|y)$ and $p(y)$ by integrating (numerically) the RHS wrt θ.

Further extensions: hidden MRFs

Consider the posterior marginal for θ. For any realisation x,

$$p(\theta|y) = \frac{p(x,\theta|y)}{p(x|\theta,y)}.$$

We can write this as

$$p(\theta|y) \propto \frac{p(y|x)p(x|\theta)p(\theta)}{p(x|\theta,y)},$$

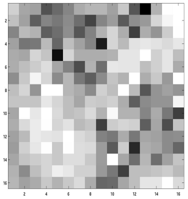
the normalising constant is $p(y)$ - the marginal likelihood.

Each term on the RHS above can be calculated exactly.

We estimate $p(\theta|y)$ and $p(y)$ by integrating (numerically) the RHS wrt θ.

We estimate these marginals *without using MCMC*

Illustrative example - moderately sized lattice

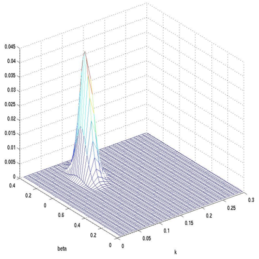


Data consist of measurements of soil phosphate content on a 16×16 grid at 10 metre intervals at a location in northern Greece (Besag, York, Mollie, 1989).

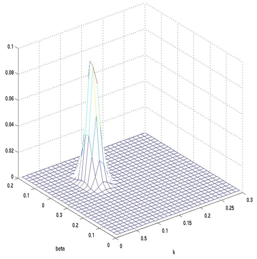
Model $k = 1$: MRF where each point has 4 nearest neighbours.
Model $k = 2$: MRF where each point has 8 nearest neighbours.

We assume y's are conditionally independent given x's with normal distribution with known means and unknown common variance κ.

Illustrative example - moderately sized lattice



$p(\theta, \kappa | y, k = 1)$



$p(\theta, \kappa | y, k = 2)$

Marginal likelihoods:

$$\log p(y|k=1) = -110.168 \text{ and } \log p(y|k=2) = -114.075$$

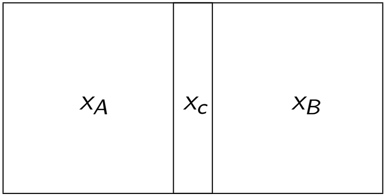
Assuming equally weighted models, a priori, yields

$$p(k=1|y) = 0.98 \text{ and } p(k=2|y) = 0.02$$

Now we ask how we can use the exact results on small lattices to do aproximate inference for larger lattices.

Large lattice approximation

Strategy: Utilise exact results on sub-lattices.

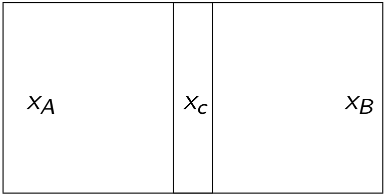


$$p(x|\theta) = p(x_A|\theta, x_c)\, p(x_c|\theta)\, p(x_B|\theta, x_c).$$

Assume we can compute both $p(x_A|\theta, x_c)$ and $p(x_B|\theta, x_c)$.

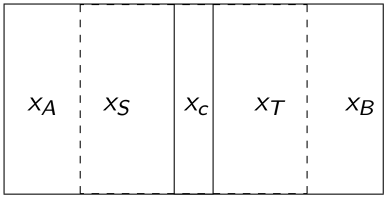
The problem remains to compute $p(x_c|\theta)$.

Large lattice approximation



$$p(x_c|\theta) = \frac{p(x|\theta)}{p(x_A|x_c,\theta)p(x_B|x_c,\theta)}$$

Large lattice approximation



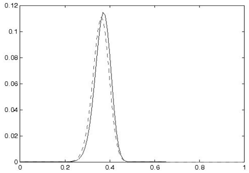
Consider a sub-lattice, $x^* = x_S \cup x_c \cup x_T$.

$$\begin{aligned} p(x_c|\theta) &= \frac{p(x|\theta)}{p(x_A|x_c,\theta)p(x_B|x_c,\theta)} \\ &\approx \frac{p(x^*|\theta)}{p(x_S|x_c,\theta)p(x_T|x_c,\theta)}. \end{aligned}$$

Large lattice approximation

Performance of the approximation

- A 19×19 realisation from an Ising model with $\theta = 0.4$ was sampled. Gaussian noise with zero mean and unit variance was added to each state value leaving data y.
- We can compute $p(\theta|y)$ very precisely, since we can compute $p(x|\theta)$ and $p(x|\theta, y)$ exactly.
- We can compare this to an estimate of $p(\theta|y)$ using the approximations to $p(x|\theta)$ and $p(x|\theta, y)$ by covering the middle column with a sub-lattice of size 19×5.

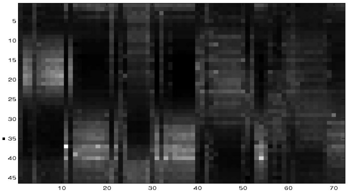


Illustrative example - larger sized lattice

Gene expressions were measured across the whole genome of *Plasmodium falciparum*, the organism that causes human malaria, for 46 1-hour consecutive intervals.

This example focuses on the relatively short mitochondrial chromosome, which consists of 72 genes and about which relatively little is known.

The data y is observed on a 46×72 spatial-temporal rectangular lattice. y_{tg} is the log-expression of gene g at time t.



Illustrative example - larger sized lattice

Latent model

The latent process is modelled a non-homogeneous Ising distribution with 2 states $\{-1, 1\}$ corresponding to 'up-regulation' and 'down-regulation'.

$$p(x|\theta) \propto \exp\left(\theta_t V_t(x) + \theta_g V_g(x)\right).$$

- $V_t(x)$ measures the interactions between neighbouring lattice points corresponding to the same gene in the 'time' direction.
- $V_g(x)$ similarly measures interactions at the same time point between neighbouring genes.

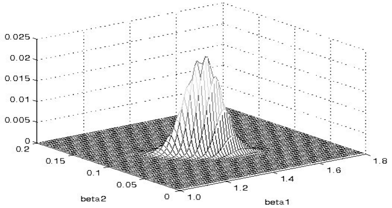
Illustrative example - larger sized lattice

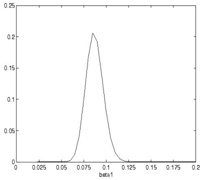
Large lattice approximation

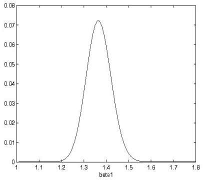
- The 46×72 lattice was partitioned into 3 disjoint sub-lattices of dimension 46×17 and a final sub-lattice of dimension 46×18, each separated by a column of lattice points.
- To compute the marginal distribution of the columns of lattice points, a lattice of size 17×46 was used to cover each column.

Illustrative example - larger sized lattice

Results







$p(\theta_t|y)$ $p(\theta_g|y)$

We now focus on some further approaches to approximate $p(\mathbf{x}|\theta)$ for large lattices $\mathbf{x}$.

Alternative large lattice approximations

Reduced dependence approximations (RDA)

Let r_i denote the ith row vector.

$$\pi(x|\theta) = \pi(r_m|\theta) \prod_{i=1}^{m-1} \pi(r_i|r_{i+1:m}, \theta).$$

We estimate each term on the RHS by conditioning on a reduced number of rows m_1.

$$\pi(x|\theta) \approx \pi(r_{m-m_1+1:m}|\theta) \prod_{i=1}^{m-m_1} \pi(r_i|r_{i+1:i+m_1}, \theta).$$

Each factor is further approximated as

$$\pi(r_i|r_{i+1:i+m_1}, \theta) \approx \frac{\pi(r_{i:i+m_1}|\theta)}{\pi(r_{i+1:i+m_1}|\theta)}.$$

Alternative large lattice approximations

Reduced dependence approximations (RDA)

$$\pi(r_i|r_{i+1:i+m_1}, \theta) \approx \frac{\pi(r_{i:i+m_1}|\theta)}{\pi(r_{i+1:i+m_1}|\theta)}.$$

Note that each probability appearing above can be calculated using the recursion method, provided $m_1 \leq 20$. In fact,

$$\pi(x|\theta) \approx \frac{q(x|\theta)}{(z_{m_1}(\theta))^{m-m_1+1}/(z_{m_1-1}(\theta))^{m-m_1}}.$$

Effectively, we approximate the overall NC as

$$z(\theta) = \frac{(z_{m_1}(\theta))^{m-m_1+1}}{(z_{m_1-1}(\theta))^{m-m_1}}.$$

Alternative large lattice approximations

Reduced dependence approximations (RDA)

$$\pi(r_i | r_{i+1:i+m_1}, \theta) \approx \frac{\pi(r_{i:i+m_1}|\theta)}{\pi(r_{i+1:i+m_1}|\theta)}.$$

Note that each probability appearing above can be calculated using the recursion method, provided $m_1 \leq 20$. In fact,

$$\pi(x|\theta) \approx \frac{q(x|\theta)}{(z_{m_1}(\theta))^{m-m_1+1}/(z_{m_1-1}(\theta))^{m-m_1}}.$$

Effectively, we approximate the overall NC as

$$z(\theta) = \frac{(z_{m_1}(\theta))^{m-m_1+1}}{(z_{m_1-1}(\theta))^{m-m_1}}.$$

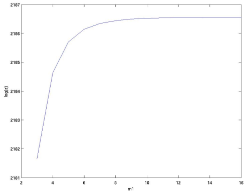
This approximation has been applied in a variational setting by McGrory et al. (2009).

Alternative large lattice approximations

Reduced dependence approximations (RDA)

What does this approximation depend on?

- The size of m_1 - the closer to m the better.
- The value of θ - the closer to 0 the better.

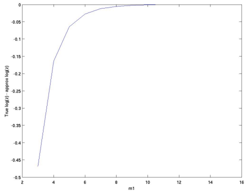


This plot displays approximations to the log NC for a 50×50 lattice with $\theta = [0, 0.4]$ for values of $m_1 = 3, \ldots, 16$.

Alternative large lattice approximations

Reduced dependence approximations (RDA)

We can compute the NC (quite fast) and exactly for a 16×16 lattice:
Here we investigate how close the approximate log NC is to the true log NC for different values of m_1



Ratio $\frac{\text{True}}{\text{Approx}} = 0.995$ for $m_1 = 8$

Alternative large lattice approximations

Partially ordered Markov model defined on sub-lattices

POMMs are a generalisation of a Markov chain to a directed acylic graph.

$$\pi(x_{ij}|x_{-ij},\theta) = \pi(x_{ij}|x_{i+1,j},x_{i,j+1},\theta)$$

Now the likelihood is tractable:

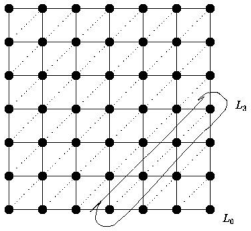
$$\pi(x|\theta) = \prod_{i=1}^{n} \pi(x_{ij}|x_{i+1,j},x_{i,j+1},\theta)$$

Alternative large lattice approximations

Partially ordered Markov model defined on sub-lattices

Alternatively, we can write the likelihood as:

$$\pi(x|\theta) = \left[\prod_{i\in L_0} \pi(x_i)\right] \left[\prod_{j=1}^{L} \prod_{i\in L_j} \pi(x_i|pa(\mathbf{x}_i))\right]$$



Now suppose that each lattice point is a sublattice for which we can compute a likelihood...

Alternative large lattice approximations

Partially ordered Markov model defined on sub-lattices

Suppose lattice x is divided into L non-overlapping sublattices $\mathbf{x}_l$.

$$\pi(\mathbf{x}_l|\theta) = \frac{1}{z_l(\theta)} \exp(\theta_0 V_0(\mathbf{x}_l) + \theta_f V_f(\mathbf{x}_l))$$

Naively we could assume independent sub-lattices,

$$\pi(\mathbf{x}|\theta) \approx \prod_{i=1}^{L} \frac{1}{z_l(\theta)} \exp(\theta_0 V_0(\mathbf{x}_l) + \theta_f V_f(\mathbf{x}_l))$$

But now dependencies across boundaries of $\mathbf{x}_l$'s have been ignored. We re-introduce these dependencies by defining a POMM with sublattices $\mathbf{x}_l$ as the nodes!

Alternative large lattice approximations

Partially ordered Markov model defined on sub-lattices

Now each sublattice $\mathbf{x}_l$ is dependent on its parent sub-lattices:

$$\pi(\mathbf{x}_l|pa(\mathbf{x}_l),\theta) = \frac{1}{z_l(\theta, pa(\mathbf{x}_l))}\exp(\theta_0 V_0(\mathbf{x}_l)+\theta_f V_f(\mathbf{x}_l)+\theta_f V_{pa}(\mathbf{x}_l, pa(\mathbf{x}_l))$$

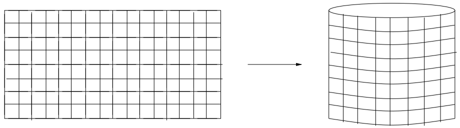
The interactions between $\mathbf{x}_l$ and it's two predecessors is taken care of by $V_{pa}(\mathbf{x}_l, pa(\mathbf{x}_l))$.
Note that the NC $z_l(\theta, pa(\mathbf{x}_l))$ is now a function of the parent sublattices.
Likelihood now looks like:

$$\pi(x|\theta) \approx \pi(\mathbf{x}_L)\prod_{l=1}^{L-1}\pi(\mathbf{x}_l|pa(\mathbf{x}_l))$$

Alternative large lattice approximations

Cylinder approximation

Assume the lattice is wrapped on a cylinder



Why is this useful?

Now every column has two neighbouring columns. Therefore, the distribution of the rows is stationary.

$$p(\mathbf{x}) = p(c_1, \ldots, c_n) = p(c_i, c_{i+1}, \ldots, c_n, c_1, \ldots, c_{i-1}),$$

where c_i is the vector of lattice points for column i.

Alternative large lattice approximations

Cylinder approximation

Let the set of all possible values of c_j be denoted by

$$A = \{a_1, \ldots, a_n\}, \text{where } \; N = 2^m.$$

Theorem

Suppose the unnormalised $q(\mathbf{x}|\theta)$ can be factorised as

$$q(\mathbf{x}|\theta) = \prod_{i=1}^{n} h(c_j, cj - 1)$$

for a given positive real function $h(\cdot, \cdot)$ defined on $A \times A$. Then the normalising constant for $q(\mathbf{x}|\theta)$ is given by $tr(Q^n)$ where Q is an $N \times N$ matrix whose kth row $(Q_{k_1}, \ldots, Q_{k_n})$ is defined by

$$h(c_1 = a_1, c_0 = a_k), h(c_1 = a_2, c_0 = a_k), \ldots, h(c_1 = a_N, c_0 = a_k)$$

for $k = 1, \ldots, N$.

Alternative large lattice approximations

Cylinder approximation

Some remarks:

1. Q is almost like a transition probability matrix. (The kth row of Q gives the probability to transition from $c_k = a_k$ to any other column).
2. $$h(c_1, c_0) = \exp\left\{\theta_0 \sum_{i=1}^{m} x_{i1} + \theta_1 \sum_{i=1}^{m-1} x_{i1}x_{i+1,1} + \theta_1 \sum_{i=1}^{m} x_{i0}x_{i1}\right\}$$
   The second term is the 'within' c_1 interactions, third term is the between c_0, c_1 interactions.
3. For a binary MRF, Q is a $2^m \times 2^m$ matrix ($m =$ no. of rows).
   $$z(\theta) = tr(Q^n) = tr(D^n),$$
   where D is the matrix of eigenvalues of Q.

Other approaches to handle intractable NCs

Recall: We are interested in the posterior

$$p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta)$$

where

$$p(\mathbf{x}|\theta) = \frac{q(\mathbf{x}|\theta)}{z(\theta)},$$

where $z(\theta)$ is an intractable normalising constant.

Approximate Bayesian Computation

First, rejection sampling:

1. $\theta \sim p(\theta)$.
2. Accept θ with probability $p(\mathbf{x}|\theta)$.

Obviously step 2 is a problem.

Approximate Bayesian Computation

However... it is often relatively easy to simulate from the model.

1. $\theta \sim p(\theta)$.
2. Simulate *pseudo-data* $\mathbf{x}^* \sim p(\cdot|\theta)$.
3. Accept θ if $s(\mathbf{x}^*) = s(\mathbf{x})$, where $s(\cdot)$ is sufficient for $p(\cdot|\theta)$.

The target distribution in this case is

$$p(\theta, \mathbf{x}^*|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta)I[s(\mathbf{x}^*) = s(\mathbf{x})]$$

The rejection ratio is

$$\frac{p(\mathbf{x}|\theta)p(\theta)I[s(\mathbf{x}^*) = s(\mathbf{x})]}{p(\mathbf{x}^*|\theta)p(\theta)} = I[s(\mathbf{x}^*) = s(\mathbf{x})]$$

Notice that it doesn't require calculation of $p(\mathbf{x}|\theta)$.

ABC is sometimes called *likelihood-free inference*.

Approximate Bayesian Computation

ABC comes with a caveat.

Simulating pseudo-data $\mathbf{x}^*$ which is similar to $\mathbf{x}$ can be difficult. Finding a sufficient statistics can also be a problem.

The ABC algorithm can be extended to augment the target even further...

$$p(\theta, \mathbf{x}^*, \epsilon | \mathbf{x}) \propto p(\mathbf{x}|\theta) p(\theta) I[d(\mathbf{x}^*, \mathbf{x}) < \epsilon].$$

This relaxes the need for $\mathbf{x}^*$ to be ‘identical’ to $\mathbf{x}$.

Other approaches

Thermodynamic integration

$$\frac{z(\theta')}{z(\theta)} = \int_{\theta}^{\theta'} \mathbf{E} \log q(\mathbf{x}|\theta^*)\ d\theta^*$$

Variational approximations

The variational approach is to propose a simple structural form for the approximation, $q(\mathbf{x}|\theta)$.

$$\hat{\theta} = \arg\min_{\theta} KL\left[p(\mathbf{x}|\theta)||q(\mathbf{x}|\theta)\right]$$

Sequential Monte Carlo

The sequential Monte Carlo approach of (Del Moral *et al.*, 2006) should be useful in these contexts also.