Appendix
for
Continuously indexed Potts model on unoriented graphs

1 Detailed derivation of the likelihood

In this section we give a more detailed derivation of the marginalization of all variables except for the extreme points out of the joint probability, on the discrete unoriented chain.

$$p(x_0, x_1, \ldots, x_l; U, h) \propto \prod_{k=0}^{l} h^\intercal x_k \prod_{k=0}^{l-1} x_k^\intercal U x_{k+1},$$

We introduce $H = \text{Diag}(h)$ and $W = H^{\frac{1}{2}} U H^{\frac{1}{2}}$, which is a parameter that combines the binary potential with half of the unary potentials from each point of an edge. We marginalize all variables except for the extreme points of the segment we then have

$$\begin{aligned}
p(x_0, x_l; W, h) &\propto \sum_{x_1 \cdots x_{l-1}} \prod_{i=0}^{l-1} x_i^\intercal U x_{i+1} \prod_{i=0}^{l} h^\intercal x_i \\
&\propto \sum_{x_1 \cdots x_{l-1}} \prod_{i=0}^{l-1} x_i^\intercal H^{-\frac{1}{2}} W H^{-\frac{1}{2}} x_{i+1} \prod_{i=0}^{l} H x_i \\
&\propto \sum_{x_1 \cdots x_{l-1}} \prod_{i=0}^{l} x_i^\intercal H^{\frac{1}{2}} \prod_{i=0}^{l-1} x_i^\intercal H^{-\frac{1}{2}} W H^{-\frac{1}{2}} x_{i+1} \prod_{i=0}^{l} H^{\frac{1}{2}} x_i \\
&\propto \sum_{x_1 \cdots x_{l-1}} x_0^\intercal H^{\frac{1}{2}} \left(\prod_{i=0}^{l-1} x_i^\intercal H^{\frac{1}{2}} x_i^\intercal H^{-\frac{1}{2}} W H^{-\frac{1}{2}} x_{i+1} H^{\frac{1}{2}} x_i \right) H^{\frac{1}{2}} x_l \\
&\propto x_0^\intercal H^{\frac{1}{2}} \left(\sum_{x_1 \cdots x_{l-1}} \prod_{i=0}^{l-1} x_i^\intercal W x_{i+1} \right) H^{\frac{1}{2}} x_l \\
&\propto x_0^\intercal H^{\frac{1}{2}} \left(x_0^\intercal W^l x_l \right) H^{\frac{1}{2}} x_l \\
&\propto h^t t x_0 \cdot x_0^\intercal H^{-\frac{1}{2}} W^l H^{-\frac{1}{2}} x_l \cdot h^\intercal x_l
\end{aligned}$$

2 Proof of lemma 1

In this section we prove the lemma regarding the neccessary and sufficient conditions on the infinestimal generator Π to ensure the term wise positivity of $\exp(\Pi * l)$, and hence all possible binary edge potential. We first prove the following lemma:

Lemma. *If Π is such that for all $i \neq j$, $\Pi_{i,j} \geq 0$, then the sequences $u_{i,j}^{(k)} = \left[\Pi^k\right]_{i,j}$ have, if it exists, a first non-zero value which is strictly positive.*

Proof. We consider the set

$$\Omega = \left\{ k \in \mathbb{N}^* | \exists\,(i,j)\,, i \neq j \mid u_{i,j}^{(k)} < 0 \text{ and } \forall t < k \Rightarrow u_{i,j}^{(t)} = 0 \right\},$$

and assume that it is non empty. It must therefore have a smallest element that we denote $k_0 = \min(\Omega)$, associated with the matrix coordinates (i_0, j_0).

By definition $u_{i_0,j_0}^{(k_0)} = \sum_{t \neq i_0} \Pi_{i_0 t} u_{t j_0}^{k_0 - 1} < 0$, which imply that the set

$$I_{i_0,j_0,k_0} = \left\{ i \neq i_0 \mid \Pi_{i_0,i} > 0,\ u_{i,j_0}^{(k_0-1)} < 0 \right\}$$

cannot be empty.

For any element i of I_{i_0,j_0,k_0} the sequence u_{i,j_0} must have taken a strictly positive value before k_0, or else we would have $\min(\Omega) \leq k_0 - 1$. This implies that the set

$$K_{i,j_0,k_0} = \left\{ k \in \mathbb{N}^* \mid k < k_0 - 1, u_{i,j_0}^{(k)} > 0 \right\}$$

is not empty.

We now consider $k_1 = \min\left(\bigcup_{i \in I_{i_0,j_0,k_0}} K_{i,j_0,k_0} \right)$, associated with i_1. Since $k_1 + 1 < k_0$, we can deduce that $u_{i_0,j_0}^{(k_1+1)} = 0$.

But the fact that $u_{i_0,j_0}^{(k_1+1)} = \sum_{t \neq i_0} \Pi_{i_0 t} u_{t j_0}^{k_1} = 0$ together with the definition of k_1 imply that $\Pi_{i_0,i} \cdot u_{i_1,j_0}^{k_1} > 0$, so that we must have that $\sum_{t \neq i_0, i_1} \Pi_{i_0 t} u_{t j_0}^{k_1} < 0$.

This implies that we can find $i_2 \neq i_0, i_1$ such that $u_{i_2 j_0}^{k_1} < 0$. But since $k_1 < k_0$ the sequence u_{i_2,j_0} must have once before taken a strictly positive value, which contradicts the definition of k_1 .

Finally, we can conclude that the set Ω is empty, which proves the lemma. □

We are now ready to prove lemma 1.

Lemma. *For Π a square matrix, $[\exp(l\,\Pi)]_{i,j} \geq 0\ \forall l \in \mathbb{R}_+$ and $\forall i, j$ if and only if $\Pi_{i,j} \geq 0$ for all $i \neq j$. Similarly, $[\exp(l\,\Pi)]_{i,j} > 0$ for all i, j and $\forall l \in \mathbb{R}_+^*$, if and only if the sequences defined as $u_{i,j}^{(k)} = \left[\Pi^k\right]_{i,j}$ is such that its first non-zero value exists and is strictly positive, for all $i \neq j$.*

Proof. We first prove the lemma for the non-strict inequalities, i.e. that if we have $\Pi_{i,j} \geq 0$ for $i \neq j$ then

$$\forall (i,j) \in \mathcal{K}^2,\ \forall l \in \mathbb{R}_+, \quad [\exp(l\,\Pi)]_{i,j} \geq 0.$$

We define the continuous functions

$$f_{i,j} : l \mapsto [\exp(l\,\Pi)]_{i,j}\,, \forall\,(i,j) \in \mathcal{K}^2,$$

and introduce the sequences of derivatives of $f_{i,j}$ at zero :

$$u_{i,j}^{(k)} = \frac{\partial^k f_{i,j}(0)}{\partial l^k} = \left[\Pi^k\right]_{i,j}.$$

Since for $i \neq j$,

$$u_{i,j}^{(k+1)} = \sum_{t=1}^{K} \Pi_{i,t} \cdot u_{t,j}^{(k)} \geq \Pi_{i,i} \cdot u_{i,j}^{(k)}$$

and $u_{i,j}^{(1)} \geq 0$, it appears that either $u_{i,j}^{(k)}$ is identically zero or its first non-null value must be strictly positive.

On the diagonal, $\forall i \in \mathcal{K}$, $f_{i,i}(0) = 1$ and since $f_{ii}(l)$ is continuous there exists an $\eta_i > 0$ such that

$$\forall s \in \mathbb{R}, |s| \leq \eta_i \Rightarrow f_{ii}(s) \geq 0.$$

Outside of the diagonal, $\forall i, j \in \mathcal{K}^2$, $f_{i,i}(0) = 0$. If the sequence $u_{i,j}$ is identically equal to zero then $f_{i,j}(0) = 0$ for all $l \in \mathbb{R}$ since $f_{i,j}(l) = \sum_{k=0}^{\infty} \frac{l^k u_{i,j}^{(k)}}{k!}$ and we set $\eta_{i,j} = 1$.

Otherwise, the first non zero derivative of $f_{i,j}$ at zero must exist and be strictly positive, implying that we can find $\eta_{i,j} > 0$ such that $\forall s \in \mathbb{R}$, $|s| \leq \eta_{i,j} \Rightarrow f_{ij}(s) \geq 0$.

With

$$\eta = \min\left((\eta_i)_{i \in K}, (\eta_{i,j})_{i,j \in \mathcal{K}^2}\right),$$

we have that $\forall l \in \mathbb{R}, \exists n \in \mathbb{N}$ such as $\left|\frac{l}{n}\right| \leq \eta$ and then, $\forall i, j \in \mathcal{K}^2$, $f_{ij}\left(\frac{l}{n}\right) \geq 0$.

The properties of the matrix exponential gives us that $[f_{ij}(l)] = \left[f_{ij}\left(\frac{l}{n}\right)\right]^n$ in term of matrix exponentiation, and since $f_{ij}\left(\frac{l}{n}\right) \geq 0 \; \forall i, j \in \mathcal{K}^2$, we also have

$$\left[f_{ij}\left(\frac{l}{n}\right)\right]^n \geq 0, \forall i, j \in \mathcal{K}^2$$

which proves that all the $f_{i,j}$ are non negative for $i \neq j$.

The proof for the strict inequality is similar because the first non zero derivatives are strictly positive, implying that the $f_{i,j}$ must be strictly positive for $t > 0$.

We now go on to prove that conversly, if $[\exp(l\,\Pi)]_{i,j} \geq 0 \; \forall l \in \mathbb{R}_+$ and $\forall i, j$ then necessarily $\Pi_{i,j} \geq 0$ for $i \neq j$.

For all $f_{i,j}$ with $i \neq j$ to be non negative the sequence of their derivatives at zero must either have their first non-zero value positive, or be identically zero, which implies that $\Pi_{i,j} \geq 0$. For the strict inequality, since we have that $[\exp(t\Pi)]_{i,j} > 0$ the sequences of $u_{i,j}^{(k)}$ cannot be identically zero. This shows that the sequences of derivatives at zero of the functions $f_{i,j}$ must have their first non-zero value and and that it must necessarily be strictly positive, which proves the lemma. □

3 Proof of lemma 5

In this seection we prove the lemma giving a compact formula for the gradient of the likelihood. We first prove the following intermediate result:

Proposition. *For x and y elementary vectors of size K, for wich the only non-zero value is set to one we have* $\nabla_\Pi \left[x^\intercal \exp(d\Pi)\, y\right] = \psi_{d,\Pi}(xy^\intercal)$, with $\psi_{d,\Pi}(X) = P^\intercal\left((PXP^\intercal) \odot \Gamma_d\right) P$ and

$$[\Gamma_d]_{i,j} = \begin{cases} \dfrac{\exp(l\,\sigma_i) - \exp(d\,\sigma_j)}{\sigma_i - \sigma_j} & \text{if } \sigma_i \neq \sigma_j \\ d\exp(d\,\sigma_j) & \text{if } \sigma_i = \sigma_j, \end{cases}$$

with $\Pi = P\mathrm{Diag}(\sigma)P^\intercal$ the eigenvalue decomposition of Π.

Proof. In the remainder of the proof, we will use the matrix max-norm defined for a matrix $M \in \mathbb{R}^{K \times K}$ by $\|M\|_{\max} = \max_{k,k'} |M_{k,k'}|$, and the matrix operator norm $\|M\|_\infty = \max_k \sum_k' |M_{k,k'}|$. We first compute the differential. For ϵ a $K \times K$ matrix such that $\|\epsilon\|_{\max} \leq 1$, we have:

$$
\begin{aligned}
& x^{\intercal}\left(\exp\left(d\ \left(\Pi+\epsilon\right)\right)\right)y - x^{\intercal}\left(\exp\left(d\,\Pi\right)\right)y \\
= \quad & x^{\intercal}\left(\sum_{k=0}^{\infty}\frac{d^k}{k!}\ \left(\left(\Pi+\epsilon\right)^k-\Pi^k\right)\right)y \\
= \quad & x^{\intercal}\left(\sum_{k=1}^{\infty}\frac{d^k}{k!}\ \left(\sum_{t=0}^{k-1}\Pi^t\epsilon\Pi^{k-1-t}+r\left(\epsilon,k\right)\right)\right)y \\
= \quad & \sum_{k=1}^{\infty}\sum_{t=0}^{k-1}\frac{d^k}{k!}\,\mathrm{Tr}\left(\epsilon\Pi^{k-1-t}yx^{\intercal}\Pi^t\right)+x^{\intercal}\sum_{k=1}^{\infty}\frac{d^k}{k!}r\left(\epsilon,k\right)y \\
= \quad & \mathrm{Tr}\left(\epsilon\left(\sum_{k=1}^{\infty}\sum_{t=0}^{k-1}\frac{d^k}{k!}\,\Pi^txy^{\intercal}\Pi^{k-1-t}\right)^{\intercal}\right)+x^{\intercal}\sum_{k=1}^{\infty}\frac{d^k}{k!}r\left(\epsilon,k\right)y,
\end{aligned}
$$

Where we have $r\left(\epsilon,k\right)$ are the terms of second order and more in the expansion of $\left(\Pi+\epsilon\right)^k$. To prove that we do have the differential we must prove that $\left|x^{\intercal}\left(\sum_{k=1}^{\infty}\frac{d^k}{k!}r\left(\epsilon,k\right)\right)y\right|$ is bound by an term which is $\mathcal{O}(\epsilon^2)$.

$$
\begin{aligned}
\left|x^{\intercal}\left(\sum_{k=1}^{\infty}\frac{d^k}{k!}r\left(\epsilon,k\right)\right)y\right| \quad &\leq \quad \sum_{k=1}^{\infty}\frac{d^k}{k!}\left\|r\left(\epsilon,k\right)\right\|_{\max} \\
&\leq \quad \left\|\epsilon\right\|_{\max}^2\sum_{k=1}^{\infty}\frac{d^k}{k!}\left\|r\left(\frac{\epsilon}{\left\|\epsilon\right\|_{\max}},k\right)\right\|_{\max} \\
&\leq \quad \left\|\epsilon\right\|_{\max}^2\sum_{k=1}^{\infty}\frac{d^k}{k!}\left\|\left(\Pi+\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\right)^k-\Pi^k-\sum_{t=0}^{k-1}\Pi^t\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\Pi^{k-1-t}\right\|_{\max} \\
&\leq \quad \left\|\epsilon\right\|_{\max}^2\sum_{k=1}^{\infty}\frac{d^k}{k!}\left(\left\|\left(\Pi+\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\right)^k\right\|_{\max}+\left\|\Pi^k\right\|_{\max}+\left\|\sum_{t=0}^{k-1}\Pi^t\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\Pi^{k-1-t}\right\|\right).
\end{aligned}
$$

We have the immediate result: $\left\|\Pi^k\right\|_{\max}\leq K^k\left\|\Pi\right\|_{\infty}$, and the less immediate one:

$$
\left\|\sum_{t=0}^{k-1}\Pi^t\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\Pi^{k-1-t}\right\|\leq kK^k\left\|\Pi\right\|_{\max}^{k-1}.
$$

Injecting those expressions in the main inequality we have that:

$$
\begin{aligned}
\left|x^{\intercal}\left(\sum_{k=1}^{\infty}\frac{d^k}{k!}r\left(\epsilon,k\right)\right)y\right| \quad &\leq \quad \left\|\epsilon\right\|_{max}^2\sum_{k=1}^{\infty}\frac{d^k}{k!}\left(K^k\left\|\Pi+\frac{\epsilon}{\left\|\epsilon\right\|_{\max}}\right\|_{\infty}^k+K^k\left\|\Pi\right\|_{\max}^k+kK^k\left\|\Pi\right\|_{\max}^{k-1}\right) \\
&\leq \quad \left\|\epsilon\right\|_{max}^2\left(\exp\left(dK\left\|\Pi\right\|_{\max}+1\right)+\left(dK+1\right)\exp\left(dK\left\|\Pi\right\|_{\max}\right)\right)
\end{aligned}
$$

This proves that:

$$
\nabla_{\Pi}\left[x^{\intercal}\exp\left(l\,\Pi\right)y\right]=\sum_{k=1}^{\infty}\sum_{t=0}^{k-1}\frac{d^k}{k!}\,\Pi^txy^{\intercal}\Pi^{k-1-t}.
$$

Since

$$
\Pi^txy^{\intercal}\Pi^{k-1-t}=P^{\intercal}\sigma^tPxy^{\intercal}P^{\intercal}\sigma^{k-1-t}P=P^{\intercal}\left(\left(Pxy^{\intercal}P^{\intercal}\right)\odot\left[\sigma_a^t\sigma_b^{k-1-t}\right]_{a,b}\right)P
$$

and

$$\sum_{t=0}^{k-1} \sigma_a^t \sigma_b^{k-1-t} = [\gamma_k]_{i,j} = \begin{cases} \dfrac{\sigma_i^k - \sigma_j^k}{\sigma_i - \sigma_j} & \text{if } \sigma_i \neq \sigma_j \\ k\sigma_i^{k-1} & \text{if } \sigma_i = \sigma_j, \end{cases}$$

we have

$$\begin{aligned} \nabla_\Pi \left[x^\intercal \exp\left(s\,\Pi\right) y\right] &= P^\intercal \sum_{k=1}^{\infty} \sum_{t=0}^{k-1} \frac{d^k}{k!} \left((Pxy^\intercal P^\intercal) \odot \left[\sigma_a^{-t} \sigma_b^{k-1-t}\right]_{a,b} \right) P \\ &= P^\intercal \left((Pxy^\intercal P^\intercal) \odot \sum_{k=1}^{\infty} \frac{d^k}{k!} \gamma_k \right) P \\ &= P^\intercal \left((Pxy^\intercal P^\intercal) \odot \Gamma_d \right) P. \end{aligned} \tag{3.1}$$

□

Proposition. $\nabla_\Pi \left[x^\intercal \Lambda(d) y\right] = \psi_d\left(xy^\intercal \oslash W^d\right)$

Proof. With Proposition 3 we have that $\nabla_\Pi \left(x^\intercal H^{-\frac{1}{2}} \exp\left(d\,\Pi\right) H^{-\frac{1}{2}} y\right) = \psi_d\left(H^{-\frac{1}{2}} xy^\intercal H^{-\frac{1}{2}}\right)$ and since

$$x^\intercal \Lambda(d) y = \log\left(x^\intercal H^{-\frac{1}{2}} W^d H^{-\frac{1}{2}} y\right)$$

we have, with $\oslash$ denoting the termwise division,

$$\nabla_\Pi \left(x^\intercal \Lambda(d) y\right) = \psi_d\left(H^{-\frac{1}{2}} \left(xy^\intercal \oslash \left(H^{-\frac{1}{2}} W^d H^{-\frac{1}{2}}\right)\right) H^{-\frac{1}{2}}\right) \tag{3.2}$$

$$= \psi_d\left(xy^\intercal \oslash W^d\right). \tag{3.3}$$

□

Proposition. $\nabla_\Pi A\left(\Pi, h\right) = \psi_d\big(\mathbb{E}\left[XY^\intercal\right] \oslash W^d\big)$

Proof. It is a classical result in the theory of exponential families that $\nabla_{\Lambda(d)} A\left(\Pi, h\right) = \mathbb{E}\left[X^\intercal Y\right]$.

By the chain rule, we have

$$\nabla_\Pi \left(A\left(\Pi, h\right)\right) = J\left(\Lambda(d), \Pi\right)^\intercal \nabla_\Pi \left[A\left(\Pi, h\right)\right].$$

where $J\left(\Lambda(d), \Pi\right)$ is the Jacobian of the function $\Pi \mapsto \Lambda(d)$, which is given by 3.3 :

$$\left[J\left(\Lambda(d), \Pi\right)\right]_{(i,j),(k,l)} = \frac{\partial \left[\Lambda(d)\right]_{(k,l)}}{\partial \Pi_{(i,j)}} = \psi_d\big(kl^\intercal \oslash W^d\big).$$

Then

$$\begin{aligned} \nabla_\Pi \left(A\left(\Pi, h\right)\right) &= J\left(\Lambda(d), \Pi\right)^\intercal \times \mathbb{E}\left[XY^\intercal\right] \\ &= \psi_d\big(\mathbb{E}\left[XY^\intercal\right] \oslash W^d\big). \end{aligned} \tag{3.4}$$

□

Subtracting the equation found in proposition 3 from the one in proposition 3 yields the gradient of the likelihood with respect to variable Π announced in lemma 6.

4 Illustration of the differences between CGMRF and CTMP

In our model the conditional probability does not only depends on the position from the conditionned variable but also from the the position in the graph. Conversly to a CTMP it is possible to have $p(x_t|x_s)$ to be non-monotonic in $t-s$ as can be seen on Figure 4.1. To obtain this figure we consider a binary process on a chain which state 1 is *repulsive*, ie transitions to state 0 are very attractive from either state, even more so than 1 to 1 transitions. As a result the probability of being in state 1 is higher on the edges of the chain than at its center because edges have only one neighbor instead of two. Even when only conditioning on the first node of the chain this gives a non-monotonic probability distribution.

To obtain Figure 4.1 we chose the following value for U and h:

$$U = \begin{bmatrix} 1.7 & .9 \\ .9 & .6 \end{bmatrix} \quad \text{and} \quad h = \begin{bmatrix} 1 \\ 1.3 \end{bmatrix}.$$

We can see that when in state 1 the process has a higher incentive to switch in state 0 in which it stays. This behavior is impossible to obtain with a homogeneous continuous time markov process.

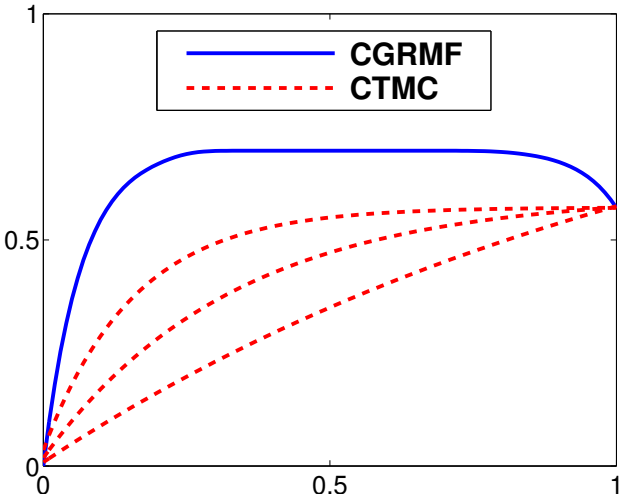

Figure 4.1: Conditional probability of being in state 0 on a segment conditionally on the first node being in state 1 for different processes: (blue) CGMRF with state 1 being *repulsive*, (red) different CTMC interpolation sharing the same conditional probability on the last node.

5 A more complete figure for transfer learning

In Figure 5.1 we see the precision-coverage curves for all the relearning settings, including the ones in which we only relearn the parameter κ. We can see that this relearning option yields results that rank in between the setting with no relearning and the full parametrization relearning.

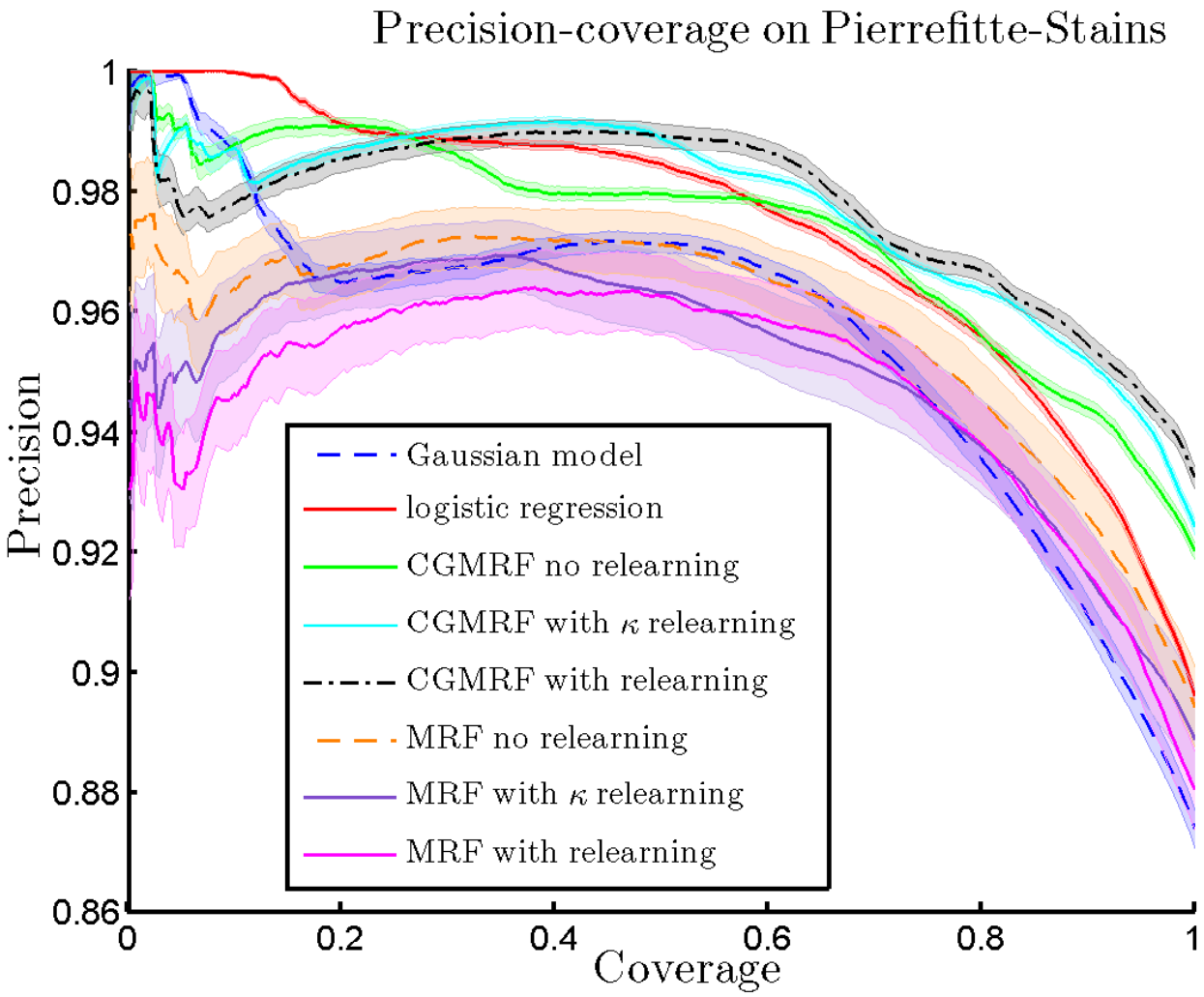

Figure 5.1: Averaged precision coverage curves for the inference on the Pierrefitte/Stains cnglomeration for 200 random sampling of 15% of revealed labels on the city of Sevran.

6 Additional experiments on artificial data

In this section we provide a detailed description of simulated experiments destined to test the core model of CGMRF in a setting with no hidden layers.

We consider two multi-class classification problems on real graphs: in the first case the real graph is a tree and the data is drawn exactly according to the model proposed in the paper, in the second case we consider the problem of predicting regions in the plane corresponding to a quantization of level sets of a random Gaussian function.

6.1 Potts process on an unoriented graph

In this first experiment, we generate a random weighted graph and generate data on this graph following the Potts model on the continuous graph. We compare the labels predicted with a CGMRF trained with the maximum likelihood principle, with the predictions obtained from the true CGMRF model, and with the prediction of a standard Potts model, i.e. which ignores the length of the edges (and which is thus the classical MRF counterpart of the CGMRF). We consider a semi-supervised setting in which only a small fraction of the labels of the nodes of the continuous graph are observed.

The graph is generated by picking greedily 3 random neighbors for each node and each edge is assigned a length sampled from a gamma distribution. Then, the variables at the junction nodes are sampled using Gibbs sampling, given a set of parameters.

We hide a portion of those variables, learn the parameters of the process following the maximum likelihood principle using a trust-region algorithm and infer the labels of the unseen nodes based those learned parameters using damped loopy belief propagation (Wainwright and Jordan, 2008, chap. 7).

For each model, we construct a precision-coverage curve reported on Figure 6.2 and based on sorting the probabilistic predictions by increasing values of the entropies of these predictions.

As a possible contender to compare our algorithm with, we consider a variant of the k-nearest neighbor (k-NN) algorithm on the graph that was called graph geodesic k-nearest neighbor in Herbster and Pontil (2006) (even though not the focus of that work) and where the geodesic distance is the shortest path in the graph in the sense of the sum of the lengths of the edges. In practice, we find nearest labelled neighbors in the geodesic sense using a simple algorithm based on a priority queue that explores recursively neighbors of neighbors. We should stress that the algorithm is not a label propagation algorithm based on the graph and that we actually follow geodesics until we find labelled points. We also consider a variant of this geodesic k-NN in which the prediction is obtained with weighted majority vote with weights that are inversely proportional to the exponential of the graph induced distance between them. The prediction is thus probabilistic and the predictor is a form of Nadaraya-Watson estimator based on the geodesic distance. For both of these methods, the number of neighbors k is chosen by cross validation. Finally we also compare with the naive algorithm which predicts systematically the most frequent label. We do not make any comparisons with graph partitioning algorithms for the reasons expressed in the discussion section.

The results are as follows. The baseline naive algorithm that constantly picks the most frequent labels of the revealed nodes attains a precision of 61% for the experiment (reported on Figure 6.2), and the geodesic k-NN algorithm cross validated on k yields a precision of 35%. For the weighted geodesic k-NN, since it produces probabilistic predictions, we report its precision-coverage curve on Figure6.2. This precision-coverage curves indicate that, even when a small proportion of node labels are revealed, the precision obtained when learning parameters is almost as high as when using the true generating distribution and significantly above the precision obtained with the discrete random Markov field. Confident predictions of our models have a much higher precision than the k-NN algorithm, which can be very useful if not all data has to be labelled or in an active learning context.

6.2 Level sets of a random Gaussian function

As mentioned in the discussion section, one of the advantages of our model over approaches based only on distance is that it can learn that some transitions between classes are more likely than others. To illustrate this we generate highly structured spatial data in the following way: we sample points uniformly on $[0, 1]^2$ and compute at each point the value of a random function obtained as a random linear combination of Gaussian functions. We then quantize these values into a finite number of classes. See Figure 6.1.

From such data, we construct a graph by connecting together the points whose Voronoi cells are adjacent. We could also have used a k nearest neighbor graph.

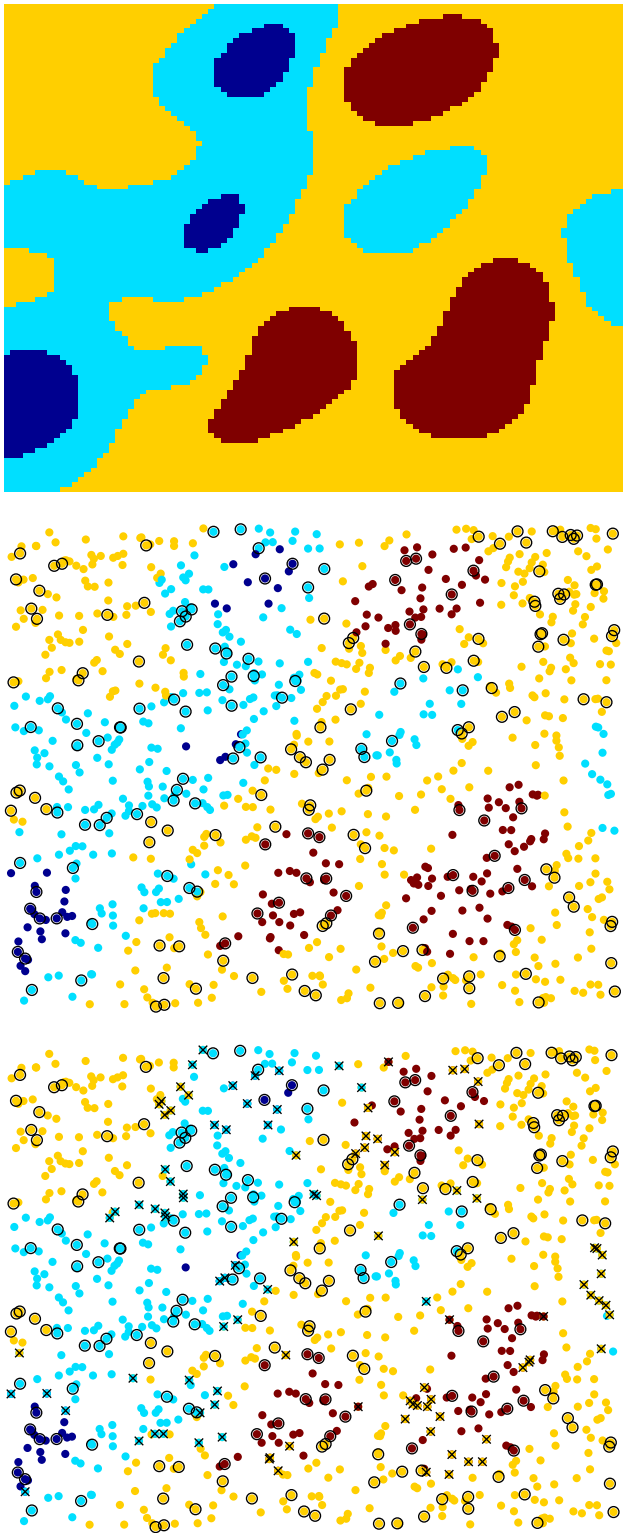

Figure 6.1: Inference of the algorithm on a random Gaussian map. (top) Quantized levels of the random Gaussian map. (middle) nodes drawn from the map with nodes whose labels are provided to the algorithm circled in black. (bottom) predictions of the CGMRF with mistakes marked with $\times$.

As a baseline we implemented a classical k-NN algorithm based on the Euclidean distance and a weighted k-NN algorithm using weights that are inversely proportional to the exponential of the Euclidean distance to the point, like in the previous experiment but with the Euclidean distance. Again k is chosen by cross validation. We compare the precision of the result of learning for standard Potts model (MRF) and for the continuous Potts model (CGMRF).

We report average precision-coverage curves over 100 replicates of the experiment on Figure 6.2.

When making prediction for all unlabeled points, the different algorithms have the following precision: for k-NN 77.4%, for the weighted k-NN 81%, the MRF 71.2%, and our CGMRF 83.5%. It is interesting to note that weighted k-NN outperforms k-NN by a large margin and that the MRF has lower precision than k-NN, even though it has a much higher precision for confident predictions. In spite of the fact that the precision coverage curve of the weighted k-NN is quite close, the misclassification error of the CGMRF is 13% smaller than weighted k-NN, 27% smaller than k-NN, and 42.7% smaller than the MRF. The gain in precision is not only obtained in average since the misclassification error of the CMRF was lower than that of its closest competitor, the weighted k-NN, in 99 out of 100 experiments, which means that the CMRF performs significantly better and by a large margin than all competitors based on Wilcoxon signed rank tests. It is interesting to note that the MRF has initially a higher precision than the CGMRF on Figure 6.2. This is explained by the fact that predictions of the CGMRF can be very confident if the closest neighbor is very close and behave in those pathological case like 1-NN, while the MRF requires that a large fraction of the neighbors have the same label to reach a similar level of confidence.

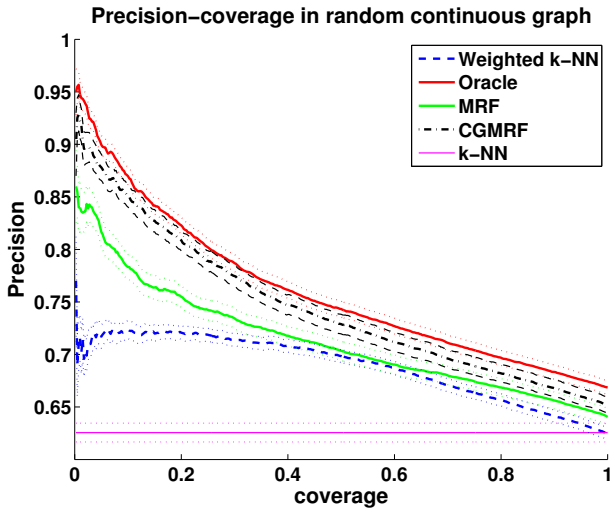

Figure 6.2: Averaged precision coverage curves for the inference in 28 random continuous trees of 500 nodes with 6 states and 20% of the labels revealed. We plot the precision of the inference with the exact parameters used to generate the data (red), parameters learnt in the continuous graph (black), in the discrete graph, or Markov random field (green), the weighted nearest neighbors alogorithm (blue) and the nearest neighbors alogorithm (magenta).

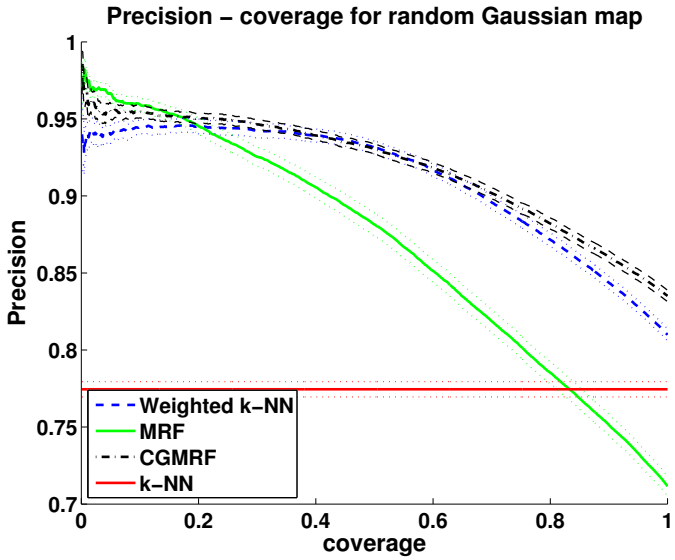

Figure 6.3: Averaged precision coverage curves for the inference in 100 random gaussian maps with 5 level sets, containing 1000 nodes each and with 20% of the labels revealed. We plot the precision of the inference parameters learnt in the continuous Potts model (black), in the Markov random Field (green), and the performance of the$k$-nearest neighbors algorithm, weighted (blue) and not (red).

References

Herbster, M. and Pontil, M. (2006). Prediction on a graph with a perceptron. In *Advances in Neural Information Processing Systems*, pages 577–584.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305.