Continuously indexed Potts models on unoriented graphs

Landrieu Loic † *
† Inria - Sierra Project-Team
École Normale Supérieure
Paris, France

Guillaume Obozinski *
* Université Paris-Est, LIGM
École des Ponts - ParisTech
Marne-la-Vallée, France

Abstract

This paper introduces an extension to undirected graphical models of the classical *continuous time* Markov chains. This model can be used to solve a transductive or unsupervised multi-class classification problem at each point of a network defined as a set of nodes connected by segments of different lengths. The classification is performed not only at the nodes, but at every point of the edge connecting two nodes. This is achieved by constructing a *Potts process* indexed by the continuum of points forming the edges of the graph.

We propose a homogeneous parameterization which satisfies Kolmogorov consistency, and show that classical inference and learning algorithms can be applied.

We then apply our model to a problem from geomatics, namely that of labelling city blocks automatically with a simple typology of classes (e.g. collective housing) from simple properties of the shape and sizes of buildings of the blocks. Our experiments shows that our model outperform standard MRFs and a discriminative model like logistic regression.

1 INTRODUCTION

Connections in networks typically have a length or weight that gives a measure of distance between the nodes connected, or the intensity of their interaction. This length information has been used to perform unsupervised or semi-supervised classification on graphs based among others on graph partitioning algorithms (see e.g. Zhu and Goldberg, 2009). When defining probabilistic graphical models on such networks, it is not clear how to take this distance into account naturally so that the interaction decreases with the distance. In this paper, we propose an unoriented counterpart of the continuous-time Markov process on a tree proposed by Holmes and Rubin (2002) which is naturally generalized to any unoriented graph.

In a continuous time Markov chain, a random state X_t is associated with every point $t \in \mathbb{R}_+$. The generalization to a continuous tree mode considered by Holmes and Rubin (2002) is most simply described through its application in phylogenetics. The phylogenetic tree of a family of species is assumed given as a directed tree with branches of different lengths. The length of the branches measure the genetic distance between extant or extinct species. Branching nodes are associated with speciation events. Each point of each branch of the tree corresponds to the form taken by a species as it existed at one time in the past and the variable modeled as a random process and defined at each such point is typically a discrete trait of that species such as the nucleic acid among $\{A, C, T, G\}$ at a certain position in the DNA. In the absence of speciation event, the state evolves like a continuous-time Markov chain, with time here being measure in terms of the genetic distance along an edge. When a branching occurs, the Markov chain is split into two identical states which continue to involve independently. For that process, if the edges of the trees are identified with line segments, there is a random variable X_t associated with every point t of each of these segments. Since a tree is simply connected, removing point t will split the tree in at least two components, and with this model we have the fundamental Markov property that the subprocesses defined on each component are conditionally independent given X_t.

We aim to extend these models in two ways. First, these continuously indexed processes are fundamentally oriented. This stems for the fact that the continuous Markov chain in this model is *homogeneous*, which implies that the conditional distributions forward in time are constant, a property which, while true forward, is not in general true backwards in time. This implies in particular that all marginals of the process on any finite set of points including at least all nodes of degree different than two is naturally parameterized as a product of conditionals $p(x_s|x_t)$ whose value depends on the graph only through the distance between s and t. We aim to propose natural parameterization for *unoriented continuously indexed models* with the same Markov property as the oriented trees. Second,

the considered models are simply connected and we would like to propose an extension from weighted trees to general weighted graphs, where all edges are identified with a real segment of lengths equal to their weights, and which satisfy the Markov property in the sense that if a finite set of points A on these segments cuts the graph into several connected components, the processes on the two subgraphs are conditionally independent given $(X_{\mathsf{a}})_{\mathsf{a}\in\mathsf{A}}$. The obtained models will be Potts models that take into account in a natural way the length of the edges and such that the interaction between two nodes decreases with the distance separating them.

After a discussion of related work, we first consider the simplest case of an unoriented continuous chain for which we propose an exponential family parametrization. Next, we show how this parametrization is naturally extended to general unoriented continuous graphs. We derive the marginal log-likelihood of different subsets of nodes, as well as the form of its gradients, and show that inference and learning in these models can be obtained with classical algorithms. We then extend the model and algorithms to the hidden Markov random field case where a feature vector is attached to a certain number node. In terms of experiments, we consider first a transductive classification problem from geomatics, which consists in assigning city blocks to different classes from simple buildings characteristics, while taking into account the distances between the blocks. Then we illustrate the possibility of using the model for transfer learning in order to refine predictions for city blocks from a new entirely unlabelled city.

2 RELATED WORK

The model we consider in this work can be viewed as an extension to undirected graphs of the continuous time Markov chain (CTMC). The continuous-time Markov chain (Norris, 1997) is a fundamental model in probability and statistics for random variables that take values in a set of discrete states and that can transition at any point in continuous time from one state to another. Beyond its theoretical value, it has been applied directly in queuing theory, for the statistical modeling of chemical reactions and in genetics.

In genetics, CTMC models have been notably used to propose models of the evolution of DNA at the nucleotide level (Nielsen, 2005; Durrett, 2008), with among several others, the celebrated Jukes-Cantor model. In this context, these models have been extended to directed trees, where the tree corresponds to a phylogeny of species or of proteins, and which has been used to estimate rate matrices or for genetic sequence alignment (Von Bing and Speed, 2004). Like for CTMCs, the fact that these models are continuous arise from temporality, and the models derived are thus intrinsically oriented. For these CTMC on trees, Holmes and Rubin (2002) proposed an exponential family parametrization of the likelihood and showed that it was possible to design an EM algorithm to learn the rate matrices modeling the substitution of DNA bases over time, in a way that generalizes the classical EM algorithm on trees.

As is the case for the CTMC, continuously indexed processes arise typically as the limit of discretely indexed processes. Along these lines, Yaple and Abrams (2013) consider a continuum limit of the Ising model on a regular grid where the lengths of the edges are infinitesimal and use it to characterize the patterns of magnetic polarity in ferromagnetic materials through the resolution of integro-differential equations.

A different but also recent line of research combining ideas from the graphical models literature with stochastic processes is known under the name of *continuous time Bayesian networks* (CTBNs, Nodelman et al., 2002). These are models of structured multivariate stochastic processes in time in which the interaction between the different components of the process can be modeled by a graphical model. These models are quite different than the continuous time tree models or the models we will propose in this paper in that, for CTBNs, the graphical model structure is somehow orthogonal to the direction of time which is the unique global oriented continuous variable for the process.

Last but not least, a common family of approaches which take into account the length of edges in a graph in the context of unsupervised or semi-supervised classification are the graph partitioning and related spectral clustering techniques (see e.g. Zhu and Goldberg, 2009, chap. 5). A review of these techniques is beyond the scope of this paper. We however discuss how these methods differ and are not directly comparable to ours in section 7.

3 NOTATIONS

All multinomial variables considered in the paper take values in $\mathcal{K} = \{1, \dots, K\}$ and are represented by the indicator vector $x \in \{0,1\}^K$ whose sole non zero entry is x_k when the multinomial is the kth state. We thus define $\mathcal{X} = \{x \in \{0,1\}^K \mid \sum_{k\in\mathcal{K}} x_k = 1\}$. Given a vector $x \in \mathbb{R}^K$, $\mathrm{Diag}(x)$ is the diagonal matrix whose elements are the entries in x.

We use $\odot$ (resp. $\oslash$) to denote the Hadamard product (resp. division), that is the entrywise multiplication (resp. division) of matrices.

We will denote nodes of graphical model with the sans-serif font a, b, and set of nodes with upper capitals of the same font: A, B.

4 CONTINUOUS GRAPH POTTS MODELS

4.1 An unoriented continuous chain model

To derive a parameterization of the model, we start with the case of an unoriented chain that we identify with the $[0, l]$

segment, where without loss of generality l is an integer. We will denote by X_{a} a multinomial random variable associated with the point $\mathsf{a} \in [0, l]$. Before defining the process at any point of the segment, we model the joint distribution of the random variables X_k for k an integer in $\{0, \dots, l\}$. Denoting by $x_k \in \{0,1\}^K$ an instance of X_k, and assuming that both unary and binary potentials are constant, the joint distribution of $(X_k)_{k\in\{0,1,\dots,l\}}$ can be written in multiplicative form as

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

with $h \in \mathbb{R}_{+*}^K$ the vector of unary potential values and $U \in \mathbb{R}_{+*}^{K\times K}$ the matrix of binary potential values. For reasons of symmetry and invariance along the chain, we assume that those parameters do not depend on the position k and that $U = U^\intercal$. Note that, while similar in spirit, the assumption that these parameters are constant is different from assuming that the Markov chain is homogeneous; we discuss this point in section 7. To get concise forms for the distributions induced on subsets of the $X_k s$ by marginalization, we introduce further $H = \mathrm{Diag}(h)$ and $W = H^{\frac{1}{2}} U H^{\frac{1}{2}}$. If in particular we marginalize all variables except for the extreme points of the segment we then have

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

(See appendix for details).
Similar calculations show that, for any sequence $\mathsf{a}_0 = 0 < \mathsf{a}_1 < \ldots < \mathsf{a}_m = l$ with $\mathsf{a}_k \in \{0, \dots, l\}$, denoting $d_j = d(\mathsf{a}_j, \mathsf{a}_{j-1}) = \mathsf{a}_j - \mathsf{a}_{j-1}$ the distances between consecutive nodes and $\mathsf{A} = \{\mathsf{a}_0, \cdots, \mathsf{a}_m\}$, we have:

$$p(x_{\mathsf{A}};\, W, h) \;\propto\; \prod_{j=0}^{m} h^\intercal x_{\mathsf{a}_j} \prod_{j=1}^{m} x_{\mathsf{a}_{j-1}}^\intercal H^{-\frac{1}{2}} W^{d_j} H^{-\frac{1}{2}} x_{\mathsf{a}_j}.$$

By simply taking the logarithm of this expression we obtain a curved exponential family of distributions with log-likelihood

$$\ell\,(x_{\mathsf{A}}; \theta) = \sum_{j=0}^{m} \eta^\intercal x_{\mathsf{a}_j} + \sum_{j=0}^{m-1} x_{\mathsf{a}_j}^\intercal \Lambda(\theta, d_j)\, x_{\mathsf{a}_{j+1}} - A(\theta), \quad (4.1)$$

with $\forall k \in \mathcal{K}$, $\eta_k = \log(h_k)$, $\theta = (W, \eta)$, A the log-partition function and where $\Lambda(\theta, d)$ is defined entrywise by $[\Lambda(\theta, d)]_{kk'} = \log([H^{-\frac{1}{2}} W^d H^{-\frac{1}{2}}]_{kk'})$.

It is now very natural to try and use this formula to extend the definition of the process to any sequence of points $\mathsf{a}_0 = 0 < \mathsf{a}_1 < \ldots < \mathsf{a}_m = l$ that are no longer restricted to take integer values. This requires however that for all $s \geq 0$, W^s should be a well defined real valued matrix with non-negative (or for learning purposes positive) entries. The fact that W is real symmetric and that all its powers should be real implies that it should have non-negative eigenvalues. Since we can approximate a low rank matrix with a full rank matrix, we assume for convenience that all it eigenvalues are positive (any low rank matrix can be approximated by a full rank one). W is then a matrix exponential $W = \exp(\Pi)$. The fact that all its powers should have non-negative entries implies in particular that for any s, W^s is *completely positive*[1]. We therefore need to characterize which conditions on Π are needed to obtain a valid W. Note that Π can be viewed as the counterpart of the rate matrix for CTMCs.

4.2 Infinitesimal generator Π

To easily compute the matrix exponential we use the eigen-decomposition of Π:

$$\Pi = P^\intercal \Sigma P, \quad \Sigma = \mathrm{Diag}(\sigma), \quad P^\intercal P = P P^\intercal = I_K \quad (4.2)$$

and exponentiate its eigenspectrum[2].

In the context of learning, it is natural to assume that the entries of W^s are actually strictly positive so that the log-likelihood is always finite. The following lemma provides sufficient and necessary conditions on Π for the entries of $\exp(l\Pi)$ to be either non negative or positive.

Lemma 1. *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 $\left(u_{i,j}^{(k)}\right)_{k\in\mathcal{N}}$ with $u_{i,j}^{(k)} = \left[\Pi^k\right]_{i,j}$ is such that its first non-zero value exists and is strictly positive, for all $i \neq j$.*

This lemma is proved in the appendix.

It is easy to see from the proof of the lemma that $\Pi_{i,j} > 0$ for $i \neq j$ is a sufficient condition for $[\exp(l\,\Pi)]_{i,j}$ to be positive for all i, j and for all $l \in \mathbb{R}_+^*$.

Note that the likelihood obtained in (4.1) is invariant by a multiplication of H or U and thus of W by a positive scalar, because of normalization. As a result it is also invariant by addition of a constant multiple of the identity matrix to Π or equivalently to σ.

This means that the likelihood is invariant by addition of an arbitrary identical constant to all the eigenvalues $(\sigma_i)_{i\in\mathcal{K}}$. In particular, it is possible to choose this constant sufficient large to guarantee that the diagonal of Π is positive. This implies that it will be conveniently possible to parameterize the model by the entrywise logarithm of Π.

[1] $A \in \mathbb{R}^{K\times K}$ is *completely positive* iff there exists $B \in \mathbb{R}_+^{K\times m}$ with $A = BB^\intercal$ (see e.g. Seber (2008) p. 223).

[2] One caveat of this parametrization is that if W is close to low rank, the corresponding eigenvalues in σ have to take large negative values. This could be addressed by working with $(\sigma_k^{-1})_{k\in\mathcal{K}}$.

4.3 Existence of the process on the chain

Proposition 2. *There exists a stochastic process* $(X_{\mathsf{a}})_{\mathsf{a}\in[0,l]}$ *defined at all points of the segment* $[0,l]$ *whose finite marginal on any finite set of points containing* a_0 *and* a_l *is given by* (4.1).

Proof. Let $\mathsf{A} = \{\mathsf{a}_0, \ldots, \mathsf{a}_m\}$ and $\mathsf{B} = \{\mathsf{b}_0, \ldots, \mathsf{b}_n\}$ two such sets with $\mathsf{a}_0 = \mathsf{b}_0 = 0$ and $\mathsf{a}_m = \mathsf{b}_n = l$. It is clear that using (4.1) to define a joint probability distribution on $(X_{\mathsf{a}})_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}}$, the distribution obtained by marginalization of elements of $\mathsf{A}\backslash\mathsf{B}$ using the same type of derivation used in (4.1) is still of the form of (4.1). Since the same holds for $\mathsf{B}\backslash\mathsf{A}$, we just showed that the collection of proposed marginals are consistent and by Kolmogorov's extension theorem (Chung and Speyer, 1998, chap. 6). This proves the existence of the process. □

4.4 Extending the model to graphs

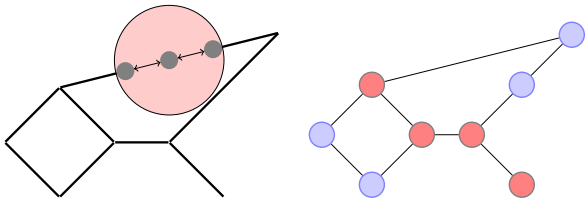
4.4.1 Real graphs

To extend the model we proposed on a segment to undirected trees and more generally to undirected graphs, we first define what we will call *continuous graphs* or *real graphs*[3]. Given a weighted graph $G = (V, E)$ with the weight d_{ab} associated with the edge $(\mathsf{a}, \mathsf{b}) \in E$, we define the associated *real graph* $\mathcal{G}$ as the space constructed as the union of line segments of lengths d_{ab} associated with the edges $(\mathsf{a}, \mathsf{b}) \in E$ and whose extreme points are respectively identified with the nodes a and b through an equivalence relation. Put informally, a real graph is the set of line segments that we usually draw to represent an abstract graph. For any pair of points a', b' on the same segment $[\mathsf{a}, \mathsf{b}]$, we will denote by $d_{\mathsf{a}'\mathsf{b}'}$ the length of that subsegment.

It should be noted that, in a real graph, the segments connecting a node of degree two are essentially merged into a single segment by concatenation. We will call all nodes of degree different than two *junction nodes*. Conversely, identifying nodes and points in the real graph, any point that is not a junction node can actually be viewed as a degree two node.

Definition 3. *Let* S *be the set of junction nodes. Given* A *a set of points on the real graph, we will call the* induced discrete graph *on* $\mathsf{A}\cup\mathsf{S}$*, denoted by* G_{A} *the graph with vertices* $\mathsf{A}\cup\mathsf{S}$ *and whose edges* E_{A} *link the nodes that can be joined on the real graph by segments not containing elements of* $\mathsf{A}\cup S$*:* $E_{\mathsf{A}} = \{(\mathsf{a}, \mathsf{b}) \mid]\,\mathsf{a}, \mathsf{b}\,[\cap (\mathsf{A}\cup\mathsf{S}) = \varnothing\}$*. To distinguish them from* $\mathsf{S}\setminus\mathsf{A}$*, we will call the set of nodes in* A observed nodes.

[3]Real graphs extend the notion of real trees which have been introduced previously in the literature (Chiswell, 2001) and are of interest notably in mathematical cladistics and to construct Brownian trees.



(a) (b)

Figure 4.1: (a) Representation of a real graph with a zoom that shows that edges are actually a continuum of nodes linked by infinestimal unoriented edges. (b) The induced discrete graph associated with the junction nodes in red and the observed nodes in blue.

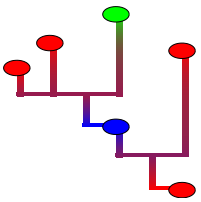


Figure 4.2: (left) Toy example illustrating that the process is defined at all points of the continuous graph. For a model on three classes (red, green blue) each point of each edge is colored with the mixture of these three colors corresponding to the probability of observing each of the classes, given that all the circle nodes are observed with the given colors.

The concepts of real graph, junction node, observed node and induced graph are illustrated on Figure 4.1.

4.4.2 Towards a Potts model on real graphs

To extend the stochastic process previously defined to real graphs, we first define its marginals. In particular, given a set of points $\mathsf{A} = \{\mathsf{a}_0, \cdots, \mathsf{a}_m\}$, the marginal on $\mathsf{A}\cup\mathsf{S}$ is naturally defined as follows: let $G_{\mathsf{A}} = (\mathsf{A}\cup\mathsf{S}, E_{\mathsf{A}})$ be the *induced discrete graph* on $\mathsf{A}\cup\mathsf{S}$, we propose to define the log-marginal distribution on $(X_{\mathsf{a}})_{\mathsf{a}\in\mathsf{A}\cup\mathsf{S}}$ as

$$\ell\,(x_{\mathsf{A}\cup\mathsf{S}};\theta) = \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{S}} \eta^{\intercal} x_{\mathsf{a}} + \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}}} x_{\mathsf{a}}^{\intercal}\Lambda(\theta, d_{\mathsf{ab}})\, x_{\mathsf{b}} - A(\theta), \quad (4.3)$$

with $\theta = (\eta, W)$ which we reparametrize from now on with $\theta = (\eta, \Pi)$. If A does not contain S, then $p\,(x_{\mathsf{A}})$ is obtained by marginalizing $x_{\mathsf{S}\setminus\mathsf{A}}$ out in $p\,(x_{\mathsf{A}\cup\mathsf{S}})$.

4.4.3 Existence of the process on a real graph

The existence of the process on a real graph is again proven using Kolmogorov's theorem:

Proposition 4. *There exists a stochastic process* $(X_{\mathsf{a}})_{\mathsf{a}\in\mathcal{G}}$ *defined at all points of the real graph* $\mathcal{G}$ *with log-marginals on any set of nodes* A *containing the junction nodes given by Eq. (4.3).*

Proof. Let A and B be two subsets of nodes on the real graph $\mathcal{G}$, for which the distributions x_{A} and x_{B} are obtained

by marginalizing S out of $x_{\mathsf{A}\cup\mathsf{S}}$ and $x_{\mathsf{B}\cup\mathsf{S}}$ in Eq. (4.3). We note that a node on an edge is conditionally independent of any node on a different edge given x_{S}. Proposition 2 tells us that the marginals are consistent on each edge with fixed endpoints, from which we can deduce that the definition of the definition of the process on $\mathsf{A}\cup\mathsf{S}$ and $\mathsf{B}\cup\mathsf{S}$ provided in Eq. (4.3) is consistent since it is obtained by marginalization of the joint distribution at the nodes $\mathsf{A}\cup\mathsf{B}\cup\mathsf{S}$. The process being consistent on A and $\mathsf{A}\cup\mathsf{S}$ by definition of $p(x_{\mathsf{A}})$, and similarly on B and $\mathsf{B}\cup\mathsf{S}$, we have proved Kolmogorov consistency between A and B which in turn proves the existence of the process on the real graph. □

We will refer to the obtained process, illustrated on Figure 4.2, as a continuous graph Potts model or continuous graph Markov random field (CGMRF).

5 INFERENCE

Probabilistic inference is an operation which is key to learning and making predictions in graphical models. In the case of our continuous graph $\mathcal{G}$, if we consider any segment $[\mathsf{a},\mathsf{b}]$ with $\mathsf{a},\mathsf{b}\in\mathsf{S}$ and any $\mathsf{a}',\mathsf{b}'\in[\mathsf{a},\mathsf{b}]$, it should be noted that $p(x_{\{\mathsf{a},\mathsf{a}',\mathsf{b}',\mathsf{b}\}}) = p(x_{\{\mathsf{a}',\mathsf{b}'\}}|x_{\{\mathsf{a},\mathsf{b}\}})p(x_{\{\mathsf{a},\mathsf{b}\}})$ where $p(x_{\{\mathsf{a},\mathsf{b}\}})$ is computed as a clique marginal of $p(x_{\mathsf{S}})$, and $p(x_{\{\mathsf{a}',\mathsf{b}'\}}|x_{\{\mathsf{a},\mathsf{b}\}})$ has a simple analytical expression given that reduces to the model on the segment. This implies that marginal distributions on any finite collection of nodes on the same edge can be computed efficiently provided the edge marginals of the induced model on S can be computed efficiently. In spite of the fact that the graph has uncountably many nodes, inference can thus be performed by any classical inference algorithm, i.e. the sum-product algorithm if the graph is a tree and typically approximate inference techniques otherwise, such as loopy belief propagation.

6 LEARNING

In this section, we focus on learning the model from data. Since the process values are only observed at a finite number of points, we are somehow always in the situation where some nodes are unobserved. However, when all junctions nodes are observed the joint likelihood of a given set of nodes has the closed form expression of Eq. (4.3). Since this a curved exponential family, the log-likelihood is in general not a concave function of the parameters[4].

To avoid having to cope with positivity constraints, and given the rapid divergence of the likelihood on the boundary of the domain we parameterize the likelihood by η and the entrywise logarithm of Π, since given the remark following lemma 1, it possible to take Π positive entrywise.

[4]It is however clearly concave when all edges are of the same length, because the constraint of equality of the parameters for all potentials is a convex constraint.

For the CTMC directed tree, Holmes and Rubin (2002) consider the likelihood of the entire process, show that it has a canonical exponential family form with a small number of sufficient statistics and derive an EM algorithm based on this representation to learn the parameters. A similar exponential family form can be obtained for our process, with also a small number of sufficient statistics and in theory it is possible to construct a similar EM algorithm. Unfortunately, in our case the M-step of the algorithm would still require solving a convex optimization problem whose solution is not closed form. We therefore do not pursue further this approach or detail the corresponding canonical exponential family form of the process. We propose instead to optimize the likelihood using a gradient based method. We show that the gradient can be computed from the moments obtained by performing the probabilistic inference on the model in different settings. In the next sections (sections 6.1 - 6.4), we derive the form of the gradient of the likelihood, first when all junction nodes are observed, then, when any set of nodes is observed, and finally, when some nodes are observed and another (typically larger) set of nodes emits observed vectors of features that are each conditionally independent given the state of associated node, as in a hidden Markov random field setting. Since computing the inference is typically intractable in graphs, we introduce a variational approximation in 6.5 that allows for faster (linear) computation. The proofs of lemmas and propositions presented can be found in the appendix.

6.1 Gradient of the likelihood on a segment

Given that the model is parameterized by exponentials of Π, the gradients involve the differential of the matrix exponential. We will therefore repeatedly use the function $\psi_{l,\Pi}$ with $\psi_{l,\Pi}(X) = P^{\intercal}\big((PXP^{\intercal})\odot\Gamma_{l,\Pi}\big)P$, where $\Pi = P\,\mathrm{Diag}(\sigma)P^{\intercal}$ is the eigenvalue decomposition of Π and

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

The function ψ is such that the gradient of $x^{\intercal}\exp(l\Pi)\,y$ is $\psi_{l,\Pi}(xy^{\intercal})$. It is essentially switching to the spectral space of Π, where the gradient has a simple multiplicative form given by Γ and then maps the result back to the original space. With this function, we thus have

Lemma 5. *The gradient with respect to variable Π of the log-likelihood ℓ of x_{a} and x_{b} on a segment of length l whose end points are a and b can be written as*

$$\nabla_{\Pi}\ell(x_{\mathsf{a}},x_{\mathsf{b}};\theta) = \psi_{l,\Pi}\big((x_{\mathsf{a}}x_{\mathsf{b}}^{\intercal}-\mathbb{E}[X_{\mathsf{a}}X_{\mathsf{b}}^{\intercal}])\oslash W^{l}\big).$$

6.2 Gradient of the likelihood in a real graph

We now compute the gradient of the log-likelihood for the joint distribution of the nodes $(x_{\mathsf{a}})_{\mathsf{a}\in\mathsf{A}}$, with the subset A

containing the junction nodes S. We still denote $G_{\mathsf{A}} = (\mathsf{A}, E_{\mathsf{A}})$ the *induced discrete graph* of A on $\mathcal{G}$. And since Π is identical for every edge, a direct application of the chain rule implies that:

Proposition 6. *The gradient of the likelihoods are computed*[5] *as*

$$\nabla_{\Pi}\ell\left(x_{\mathsf{A}};\theta\right) = \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}}} \psi_{d_{\mathsf{ab}},\Pi}\big(\left(x_{\mathsf{a}}x_{\mathsf{b}}^{\intercal} - \mu_{\mathsf{ab}}\right) \oslash W^{d_{\mathsf{ab}}}\big)$$

$$\nabla_{\eta}\ell\left(x_{\mathsf{A}};\theta\right) = \sum_{\mathsf{a}\in\mathsf{A}} \left(x_{\mathsf{a}} - \mu_{\mathsf{a}}\right) - \tfrac{1}{2}\sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}}} \left(x_{\mathsf{a}} - \mu_{\mathsf{a}} + x_{\mathsf{b}} - \mu_{\mathsf{b}}\right).$$

with $\mu_{\mathsf{a}} = \mathbb{E}\left[X_{\mathsf{a}}\right]$ and $\mu_{\mathsf{ab}} = \mathbb{E}\left[X_{\mathsf{a}}X_{\mathsf{b}}^{\intercal}\right]$.

6.3 Partially observed junction nodes

To learn from partially labelled data it is necessary to consider the likelihood of X_{B} for B a set of nodes that does not necessarily contain S. Let B be a set of observed nodes, i.e. for which we know the states x_{B}, and A a set of unobserved nodes containing $\mathsf{S}\backslash\mathsf{B}$. We have the following distributions

$$\ell\left(x_{\mathsf{A}\cup\mathsf{B}};\theta\right) = \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}} \eta^{\intercal}x_{\mathsf{a}} + \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}\cup\mathsf{B}}} x_{\mathsf{a}}^{\intercal}\Lambda(\theta, d_{\mathsf{ab}})x_{\mathsf{b}} - A_{\mathsf{A}\cup\mathsf{B}}(\theta)$$

$$\ell\left(x_{\mathsf{A}}|x_{\mathsf{B}};\theta\right) = \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}} \eta^{\intercal}x_{\mathsf{a}} + \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}\cup\mathsf{B}}} x_{\mathsf{a}}^{\intercal}\Lambda(\theta, d_{\mathsf{ab}})x_{\mathsf{b}} - A_{\mathsf{A}|\mathsf{B}}(\theta, x_{\mathsf{B}})$$

We can rewrite the log-likelihood as follows (Wainwright and Jordan, 2008) :

$$\ell\left(x_{\mathsf{B}};\theta\right) = A_{\mathsf{A}|\mathsf{B}}\left(\Pi, h, x_{\mathsf{B}}\right) - A_{\mathsf{A}\cup\mathsf{B}}\left(\Pi, h\right),$$

and its gradient are therefore computed as

Proposition 7.

$$\nabla_{\Pi}\ell\left(x_{\mathsf{B}};\theta\right) = \sum_{\langle\mathsf{a},\mathsf{b}\rangle\in E_{\mathsf{A}\cup\mathsf{B}}} \psi_{d_{\mathsf{ab}},\Pi}\left(\left(\mu_{\mathsf{ab}|\mathsf{B}} - \mu_{\mathsf{ab}}\right) \oslash W^{d_{\mathsf{ab}}}\right)$$

$$\nabla_{\eta}\ell\left(x_{\mathsf{B}};\theta\right) = \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}} \mu_{\mathsf{a}|\mathsf{B}} - \mu_{\mathsf{a}} - \tfrac{1}{2}\sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}\cup\mathsf{B}}} \left(\mu_{\mathsf{a}|\mathsf{B}} - \mu_{\mathsf{a}} + \mu_{\mathsf{b}|\mathsf{B}} - \mu_{\mathsf{b}}\right).$$

with $\mu_{\mathsf{ab}|\mathsf{B}} = \mathbb{E}\left[X_{\mathsf{a}}X_{\mathsf{b}}^{\intercal}|X_{\mathsf{B}} = x_{\mathsf{B}}\right]$, $\mu_{\mathsf{a}|\mathsf{B}} = \mathbb{E}\left[X_{\mathsf{a}}|X_{\mathsf{B}} = x_{\mathsf{B}}\right]$.

6.4 Hidden Markov model

We consider a hidden Markov random field variant of our model in which some nodes have, in addition to the state variable, a feature vector with a state specific distribution. More precisely, we envision to learn from data on a graph in which the states of a set of nodes B are observed and in which each node in a set A (with $\mathsf{A}\cap\mathsf{B} \neq \varnothing$) provides an observed feature vectors y_{a} which is conditionally independent of the rest of the graph given the corresponding node state x_{a}. For simplicity, we assume that $\mathsf{S} \subset \mathsf{A}\cup\mathsf{B}$.

The joint and conditional distribution of observed and unobserved variables are very similar as above

$$\begin{aligned}\ell\left(x_{\mathsf{A}\cup\mathsf{B}}, y_{\mathsf{A}};\theta,\kappa\right) &= \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}} \eta^{\intercal}x_{\mathsf{a}} + \sum_{\mathsf{a}\in\mathsf{A}} \log\left(p\left(y_{\mathsf{a}}|x_{\mathsf{a}};\kappa\right)\right) \\ &\quad + \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}\cup\mathsf{B}}} x_{\mathsf{a}}^{\intercal}\Lambda(\theta, d_{\mathsf{ab}})x_{\mathsf{b}} - A_{\mathsf{A}\cup\mathsf{B}}\left(\theta,\kappa\right) \\ \ell\left(x_{\mathsf{A}}|y_{\mathsf{A}}, x_{\mathsf{B}};\theta,\kappa\right) &= \sum_{\mathsf{a}\in\mathsf{A}\cup\mathsf{B}} \eta^{\intercal}x_{\mathsf{a}} + \sum_{\mathsf{a}\in\mathsf{A}} \log\left(p\left(y_{\mathsf{a}}|x_{\mathsf{a}}\right);\kappa\right) \\ &\quad + \sum_{(\mathsf{a},\mathsf{b})\in E_{\mathsf{A}\cup\mathsf{B}}} x_{\mathsf{a}}^{\intercal}\Lambda(\theta, d_{\mathsf{ab}})x_{\mathsf{b}} - A_{\mathsf{A}|\mathsf{B}}\left(\theta,\kappa, x_{\mathsf{B}}, y_{\mathsf{A}}\right),\end{aligned}$$

which allows us to rewrite the likelihood of observations as $\ell\left(x_{\mathsf{B}}, y_{\mathsf{A}}\right) = A_{\mathsf{A}|\mathsf{B}}\left(\theta,\kappa, y_{\mathsf{A}}, x_{\mathsf{B}}\right) - A_{\mathsf{A}\cup\mathsf{B}}\left(\theta,\kappa\right)$.
Given that the model for $p(y_{\mathsf{a}}|x_{\mathsf{a}})$ is Gaussian or at least an exponential family, when envisioning an EM algorithm to learn κ and θ, it is easy to see that the update for κ is closed form while that of θ is not. This motivates a variant of the EM algorithm which does not attempt to maximize with respect to both κ and θ simultaneously but which either maximizes the expected likelihood with respect to κ or maximizes it with respect to θ. The algorithm can then be summarized as an E-M1-E-M2 algorithm, where the E-step is the usual computation of expected sufficient statistics given current parameters, M1 solves for κ in closed form and M2 maximizes with respect to θ using gradient ascent[6].

6.5 Variational approximation

For graphs with cycles, since inference is intractable, we replace the likelihood by a pseudo-likelihood obtained using a variational approximation of the log-partition. Our variational approximation is the one associated with the entropy of Bethe (see, e.g. section 4.1 in Wainwright and Jordan, 2008), but other choices would be possible. The main motivation behind this approximation is that the exact gradient of this pseudo-likelihood is directly obtained from the pseudo-moments given by loopy BP. In practice, damping needs to be used (see Wainwright and Jordan, 2008, chap. 7).

In term of complexity, the parametrization of CGMRF could suggest that inference is slower than in the discrete setting since the computation of the SVD of Π is required. However, since the number of states is typically much smaller than the number of nodes in the graph, the computational cost of the SVD is negligeable compared to

[5] Note that a single spectral decomposition of Π allows to compute $W^{d_{\mathsf{ab}}}$ efficiently for all pairs (a,b).

[6] Note that gradient ascent itself requires to perform some inference to recompute the log-partition function

the overall cost of the algorithm. Hence, inference in the CGMRF is just as hard as for any discrete MRF.

The log-likelihood is a curved exponential family and is in particular not a convex function of the parameters, while it is convex for a standard MRF. As a consequence the pseudo log-likelihood based on the variational approximation is also non-convex. We use gradient descent with a line-search based on the Wolfe conditions to find a local minimum (see Nocedal and Wright, 1999, chap. 3). Empirically the algorithm is not trapped in bad local minima but takes more iterations to converge than the MRF counterpart. Experiments showed that the training for CGMRFs was only two times longer than for regular MRFs.

7 DISCUSSION

In this section, we discuss more precisely features of CGMRFs that are unique or common with other models and approaches existing in the literature.

First, we note that for a tree, our model is not equivalent to that of Holmes and Rubin (2002). Their model uses a constant rate matrix (i.e. the Markov process is homogeneous) while we use constant infinitesimal potentials, which do not lead to a constant rate matrix on any orientation of the tree. If the tree is just the segment $[0, L]$, for s and t with $0 < s < t < L$ a CTMC is such that $p(x_t|x_s)$ only depends on $t - s$ and not on L. By contrast for our model $\log p\left(x_t|x_s\right)$ depends also on $L - t$ and $L - s$ since $\log p\left(x_t|x_s\right) = x_s^\intercal \Lambda\left(t-s\right) x_t + x_t^\intercal \eta + x_t^\intercal \Lambda\left(L-t\right)\mathbf{1} - x_s^\intercal \Lambda\left(L-s\right)\mathbf{1}$, where for simplicity we omitted the dependance in θ, and $\mathbf{1}$ is the constant vector equal to 1. See the appendix for an illustration and further discussion of the differences between the models.

Our model has in common with graph partitioning techniques and spectral clustering (SC) that the distance between nodes are taken into account. But there are several important differences: first, in SC, there is no model learning in the sense that no parameters are learned to optimize the model (Bach and Jordan (2006) who learn the metric for SC, are an exception). Second, our model captures that there could be different transition probabilities between different classes along the graph which is not possible in SC. Then, the main assumption in SC is that classes are separated by edges of smaller weights so that each class is as disconnected as possible. By contrast, our model authorizes (to some extent) transitions between classes on short edges and moreover permits that each class corresponds to several connected components. Our models extends naturally to a hidden Markov model that makes it possible to include feature vectors for some nodes and not for others, which is not possible with SC techniques.

Another graph-based approach to classification which is perhaps more related to ours is the work of Zhu et al. (2003) on binary classification with harmonic functions. Indeed, the Gaussian field considered there is similar to the Potts model we obtain on the junction nodes. The approach of Zhu et al. (2003) is however just concerned by inference and not by learning, but their approach could be extended both to multi-class classification and to perform learning of the parameters.

8 EXPERIMENTS

We present in this section experiments on real data. Synthetic experiments on the core model of the CGMRF (without hidden layer) can be found in section 6 of the appendix.

In geographic information systems, data is often aggregated either on regular grid or on cells corresponding to abstract administrative boundaries, which do not necessarily reflect the structure of a city. A fairly natural type of representation for urban environment is based on graphs and in particular weighted graphs which can encode a distance information.

We consider a problem from geomatics in which this type of representation could be beneficial and which consists in predicting building use in urban and peri-urban environments from a few annotations and simple building shape characteristics that can be extracted easily from aerial images. More precisely, we consider the transductive learning problem of assigning city blocks to one category from {*individual housing, collective housing, industrial/commercial area*}.

8.1 Building the city block continuous graph

A city can be divided into city blocks using its layout and road network as in Figure 8.1. Assuming that the blocks are given, we compute the Voronoi diagram of the block centroids and link together blocks with adjacent Voronoi cells. Edges are annotated with a proximity measure, in our case the distance between their respective closest buildings. This provides a continuous graph encapsulating the s tructure of the city. Each block is then annotated into one of three categories : individual residential, collective residential and industrial/commercial area. The blocks are annotated by hand using cadastral information, business registration codes, and resorting to Google street view images for ambiguous blocks (see Figure 8.1).

8.2 Data descriptors and learning setting

A block is then described by the weighted average of characteristics of the buildings it contains, each building counting with a weight proportional to its volume. We tested 10 different building descriptors, found that floor area and height were the most discriminative, and that adding more descriptors actually decreases the performance of all tested algorithms.

We use the example of Sevran, a French city of 50 000 inhabitants north of Paris. We divided it into 461 blocks,

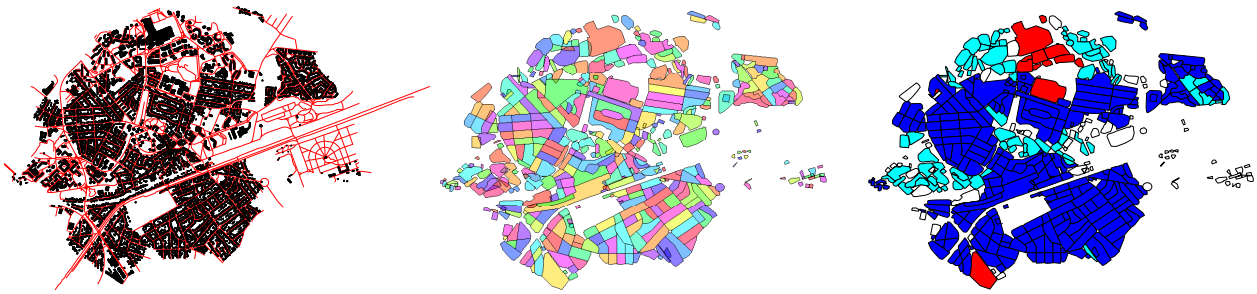


Figure 8.1: (left) Buildings and road network of Sevran. (middle) Division into city blocks. (right) City blocks with annotations. Blue: individual housing, cyan: collective housing, red: industrial/commercial area. (Best seen in color.)

400 of which can clearly be assigned one of three labels mentioned above and the rest being of insignificant size, ambiguous, or corresponding to other categories such as schools or hospitals.

We consider the transductive learning problem of predicting all block labels from a subset of labelled blocks. In our experiments, 7% of annotated labels, corresponding to 28 blocks, are used for training and the remaining are used for testing.

8.3 Competing algorithms

As baselines we consider two algorithms that do not take into account spatial information: a generative Gaussian mixture model and a logistic regression trained each using the 7% revealed labels. We also consider classical hidden MRFs, which cannot take into account the distance, and whose graph is either the same as for the CGMRF or a pruned graph in which all edges longer than a threshold (corresponding to the average city block radius) have been removed. The different graphs are illustrated on Figure 8.2. Note that the Gaussian mixture model does not take the graph structure into account, and can be interpreted as an edgeless MRF

In all Markov models, we use Gaussian emissions to model the distribution of the building descriptors given the block label, which can conveniently be optimized in closed form. To train the CGMRF and MRF models we learn the parameter θ with the maximum likelihood principle following the approach presented in section 6.5.

8.4 Results analysis

For each model, we construct a precision-coverage curve, obtained by sorting the probabilistic predictions by increasing values of their entropies, and reported on Figure 8.3. The confidence bands represented corresponds to one standard error for the estimation of the mean precision.

We can see that enriching the simple Gaussian mixture model by adding a graph structure significantly improves the overall performance. Building a MRF using all the edges from the Voronoi proximity or only retaining a fraction of the shorter edges yields similar results, on par with logistic regression. Building a CMRF using the edges annotated with a distance measure leads to a performance which is significantly above all others based on estimated standard errors.

When making prediction for all unlabeled points from the 7% of revealed annotations, the different algorithms yield the following average precisions (over the 300 resamplings): for the Gaussian mixture model 88.0%, for logistic regression 92.5%, the full MRF 92.4%, the pruned MRF 91.6% and our CGMRF 94.0%. Both pruned MRF and full MRF outperform the simple Gaussian mixture model, but not logistic regression, even though their precision at intermediate coverage is higher. The misclassification error of the CGMRF is 20% smaller than that of logistic regression, 21.5% smaller than for the best MRF model, and 50.2% smaller than for the Gaussian mixture. The gain in precision is not only obtained in average since the misclassification error in the CMRF was lower than MRF and logistic regression in respectively 193 and 293 out of 300 experiments. Wilcoxon signed rank tests assigns respectively p-values of $7 \cdot 10^{-26}$ and $3 \cdot 10^{-24}$ to the common median hypothesis.

In this experiment, with 461 nodes and 2718 edges the inference takes less than 0.1s on a CPU at 3.3GHz. Learning requires usually around 50 calls to the inference step for the MRF (5s total), while it is closer to 100 for the CGMRF (10s total).

8.5 Transfer learning on another city

We now consider the problem of predicting block labels on a new unannotated city using partial annotation from a given city. More precisely, we train our model with 15% of revealed labels from Sevran, and consider several

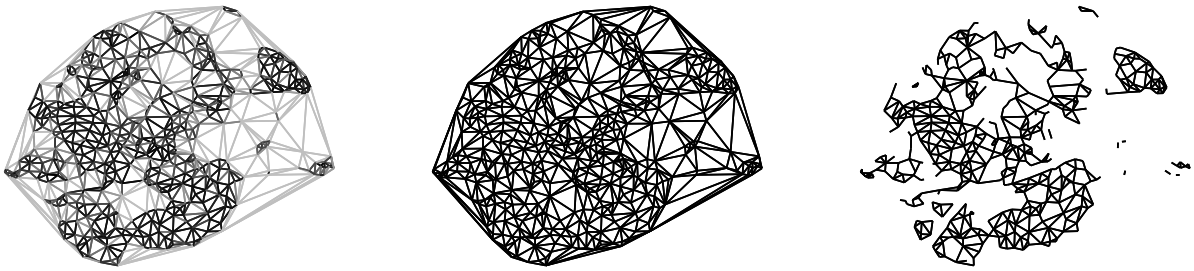


Figure 8.2: (left) continuous graph used to train the HCGMRF, the darker the edge the shorter the annoted distance, (middle) graph used for the HMRF including all edges or (right) with only edges shorter than a threshold.

schemes to make predictions on the neighboring urban area formed by Pierrefitte-sur-Seine together with Stains, for a total of 63000 inhabitants and 583 blocks, for which both graph and features are available but no labels are revealed. We consider logistic regression and the Gaussian mixture model trained from the annotated blocks from Sevran as baselines, and test for each of the CGMRF and MRF the models learnt as follows:

- θ and κ are learnt on data from Sevran
- idem followed by a single EM-step on κ alone (E-M2) on the graph of Pierrefitte+Stains
- idem followed by an EM-step on θ (E-M1) and then an EM-step on κ (E-M2).

We use the 359 labelled blocks (out of 583) of the Pierrefitte/Stains conglomeration as a testing set and construct the precision-coverage curves reported on Figure 8.4 (see the appendix for a figure comaring more approaches). We observe that the CGMRF setting is superior to its competitors, and that the relearning step improves the performance. The MRFs does not perform as well, which can be explained by the initial prediction being inferior, and relearning degrades its performance. The setting where only one E-M2 step is performed yields in both cases results comprised between the two other settings.

9 CONCLUSION

In this paper, we constructed a Potts model over a continuous graph and showed how to compute the likelihood of several of its variants as well as the corresponding gradients, for the purpose of learning.

Our experiments on a problem from geomatics show that this model outperforms regular MRFs, and compares favorably with logistic regression which although discriminative does not leverage unlabelled data. Finally, we showed that the model can be used to perform transfer learning from a first partially labelled graph towards a new completely unlabeled graph.

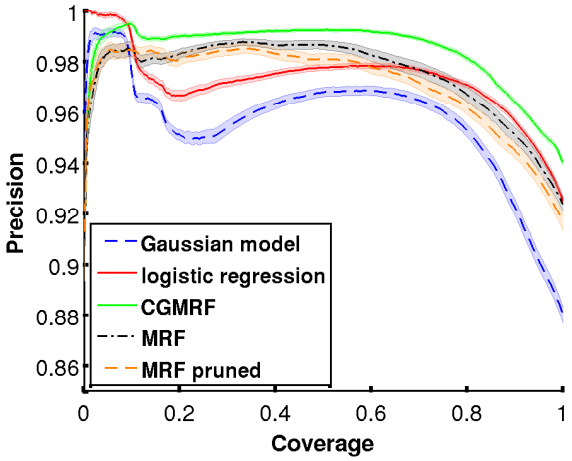


Figure 8.3: **Precision coverage curves on Sevran.** Averaged precision coverage curves for the inference for 300 random resamplings of 7% of revealed labels on the city of Sevran. (Best seen in color.)

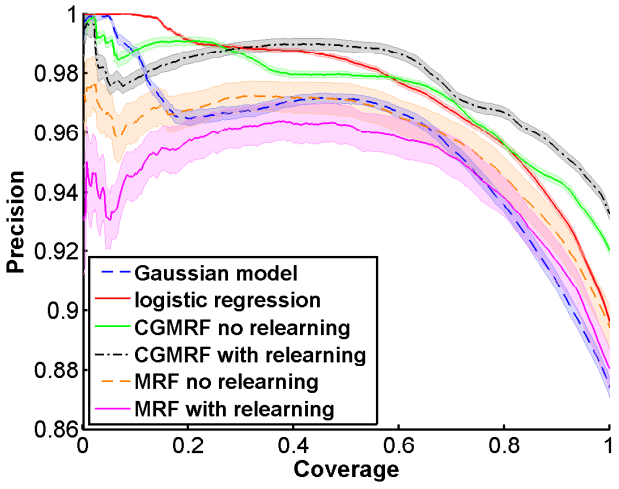


Figure 8.4: **Precision coverage curves for transfer learning.** Averaged precision coverage curves for the inference on the Pierrefitte/Stains conglomeration for 200 random resamplings of 15% of revealed labels on the city of Sevran. (Best seen in color.)

References

Bach, F. R. and Jordan, M. I. (2006). Learning spectral clustering, with application to speech separation. *The Journal of Machine Learning Research*, 7:1963–2001.

Chiswell, I. (2001). *Introduction to Lambda Trees*. World Scientific Publishing Company.

Chung, W. H. and Speyer, J. L. (1998). *Stochastic Processes, Estimation, and Control*. Society for Industrial and Applied Mathematics.

Durrett, R. (2008). *Probability models for DNA sequence evolution*. Springer.

Holmes, I. and Rubin, G. (2002). An expectation maximization algorithm for training hidden substitution models. *Journal of Molecular Biology*, 317(5):753–764.

Nielsen, R. (2005). *Statistical methods in molecular evolution*. Springer.

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer.

Nodelman, U., Shelton, C. R., and Koller, D. (2002). Continuous time Bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 378–387. Morgan Kaufmann Publishers Inc.

Norris, J. R. (1997). *Markov chains*. Cambridge University Press.

Seber, G. A. (2008). *A matrix handbook for statisticians*, volume 15. Wiley.

Von Bing, Y. and Speed, T. P. (2004). Modeling DNA base substitution in large genomic regions from two organisms. *Journal of Molecular Evolution*, 58(1):12–18.

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.

Yaple, H. A. and Abrams, D. M. (2013). A continuum generalization of the Ising model. *arXiv1306.3528*.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 3, pages 912–919.

Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130.