Learning Heterogeneous Hidden Markov Random Fields

Jie Liu
CS, UW-Madison

Chunming Zhang
Statistics, UW-Madison

Elizabeth Burnside
Radiology, UW-Madison

David Page
BMI & CS, UW-Madison

Abstract

Hidden Markov random fields (HMRFs) are conventionally assumed to be homogeneous in the sense that the potential functions are invariant across different sites. However in some biological applications, it is desirable to make HMRFs heterogeneous, especially when there exists some background knowledge about how the potential functions vary. We formally define heterogeneous HMRFs and propose an EM algorithm whose M-step combines a contrastive divergence learner with a kernel smoothing step to incorporate the background knowledge. Simulations show that our algorithm is effective for learning heterogeneous HMRFs and outperforms alternative binning methods. We learn a heterogeneous HMRF in a real-world study.

1 Introduction

Hidden Markov models (HMMs) and hidden Markov random fields (HMRFs) are useful approaches for modelling structured data such as speech, text, vision and biological data. HMMs and HMRFs have been extended in many ways, such as the infinite models [Beal et al., 2002, Gael et al., 2008, Chatzis and Tsechpenakis, 2009], the factorial models [Ghahramani and Jordan, 1997, Kim and Zabih, 2002], the high-order models [Lan et al., 2006] and the nonparametric models [Hsu et al., 2009, Song et al., 2010]. HMMs are homogeneous in the sense that the transition matrix stays the same across different sites. HMRFs, intensively used in image segmentation tasks [Zhang et al., 2001, Celeux et al., 2003, Chatzis and Varvarigou, 2008], are also homogeneous. The homogeneity assumption for HMRFs in image segmentation tasks is legitimate, because people usually assume that the neighborhood system on an image is invariant across different regions. However, it is necessary to bring heterogeneity to HMMs and HMRFs in some biological applications where the correlation structure can change over different sites. For example, a heterogeneous HMM is used for segmenting array CGH data [Marioni et al., 2006], and the transition matrix depends on some background knowledge, i.e. some distance measurement which changes over the sites. A heterogeneous HMRF is used to filter SNPs in genome-wide association studies [Liu et al., 2012a], and the pairwise potential functions depend on some background knowledge, i.e. some correlation measure between the SNPs which can be different between different pairs. In both of these applications, the transition matrix and the pairwise potential functions are heterogeneous and are parameterized as monotone parametric functions of the background knowledge. Although the algorithms tune the parameters in the monotone functions, there is no justification that the parameterization of the monotone functions is correct. *Can we adopt the background knowledge about these heterogeneous parameters adaptively during HMRF learning, and recover the relation between the parameters and the background knowledge nonparametrically?*

Our paper is the first to learn HMRFs with heterogeneous parameters by adaptively incorporating the background knowledge. It is an EM algorithm whose M-step combines a contrastive divergence style learner with a kernel smoothing step to incorporate the background knowledge. Details about our EM-kernel-PCD algorithm are given in Section 3 after we formally define heterogeneous HMRFs in Section 2. Simulations in Section 4 show that our EM-kernel-PCD algorithm is effective for learning heterogeneous HMRFs and outperforms alternative methods. In Section 5, we learn a heterogeneous HMRF in a real-world genome-wide association study. We conclude in Section 6.

2 Models

2.1 HMRFs And Homogeneity Assumption

Suppose that $\mathbb{X} = \{0, 1, ..., m-1\}$ is a discrete space, and we have a *Markov random field* (MRF) defined on

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

a random vector $\mathbf{X} \in \mathbb{X}^d$. The conditional independence is described by an undirected graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$. The node set $\mathbb{V}$ consists of d nodes. The edge set $\mathbb{E}$ consists of r edges. The probability of $\mathbf{x}$ from the MRF with parameters $\boldsymbol{\theta}$ is

$$P(\mathbf{x};\boldsymbol{\theta}) = \frac{Q(\mathbf{x};\boldsymbol{\theta})}{Z(\boldsymbol{\theta})} = \frac{1}{Z(\boldsymbol{\theta})} \prod_{c\in\mathbb{C}(\mathbb{G})} \phi_c(\mathbf{x};\boldsymbol{\theta}_c), \quad (1)$$

where $Z(\boldsymbol{\theta})$ is the normalizing constant. $Q(\mathbf{x};\boldsymbol{\theta})$ is some unnormalized measure with $\mathbb{C}(\mathbb{G})$ being some subset of the cliques in $\mathbb{G}$. The potential function ϕ_c is defined on the clique c and is parameterized by $\boldsymbol{\theta}_c$. For simplicity in this paper, we consider pairwise MRFs, whose potential functions are defined on the edges, namely $|\mathbb{C}(\mathbb{G})| = r$. We further assume that each pairwise potential function is parameterized by a single parameter, i.e. $\boldsymbol{\theta}_c = \{\theta_c\}$.

A *hidden Markov random field* [Zhang et al., 2001, Celeux et al., 2003, Chatzis and Varvarigou, 2008] consists of a hidden random field $\mathbf{X} \in \mathbb{X}^d$ and an observable random field $\mathbf{Y} \in \mathbb{Y}^d$ where $\mathbb{Y}$ is another space (either continuous or discrete). The random field $\mathbf{X}$ is a Markov random field with density $P(\mathbf{x};\boldsymbol{\theta})$, as defined in Formula (1), and its instantiation $\mathbf{x}$ cannot be measured directly. Instead, we can observe the emitted random field $\mathbf{Y}$ with its individual dimension Y_i depending on X_i for $i = 1, ..., d$, namely $P(\mathbf{y}|\mathbf{x};\boldsymbol{\varphi}) = \prod_{i=1}^d P(y_i|x_i;\boldsymbol{\varphi})$ where $\boldsymbol{\varphi} = \{\varphi_0, ..., \varphi_{m-1}\}$ and φ_{x_i} parameterizes the emitting distribution of Y_i under the state x_i. Therefore, the joint probablity of $\mathbf{x}$ and $\mathbf{y}$ is

$$\begin{aligned} P(\mathbf{x},\mathbf{y};\boldsymbol{\theta},\boldsymbol{\varphi}) &= P(\mathbf{x};\boldsymbol{\theta})P(\mathbf{y}|\mathbf{x};\boldsymbol{\varphi}) \\ &= \frac{1}{Z(\boldsymbol{\theta})} \prod_{c\in\mathbb{C}(\mathbb{G})} \phi_c(\mathbf{x};\boldsymbol{\theta}_c) \prod_{i=1}^d P(y_i|x_i;\boldsymbol{\varphi}). \end{aligned} \quad (2)$$

Example 1: One pairwise HMRF model with three latent variables (X_1, X_2, X_3) and three observable variables (Y_1, Y_2, Y_3) is given in Figure 1. Let $\mathbb{X} = \{0, 1\}$. X_1, X_2 and X_3 are connected by three edges. The pairwise potential function ϕ_i on edge i (connecting X_u and X_v) parameterized by θ_i ($0<\theta_i<1$) is $\phi_i(\mathbf{X};\theta_i) = \theta_i^{I(X_u=X_v)}(1-\theta_i)^{I(X_u\neq X_v)}$ for $i = 1, 2, 3$, where I is an indicator variable. Let $\mathbb{Y} = R$. For $i = 1, 2, 3$, $Y_i|X_i{=}0 \sim N(\mu_0, \sigma_0^2)$ and $Y_i|X_i{=}1 \sim N(\mu_1, \sigma_1^2)$, namely $\varphi_0 = \{\mu_0, \sigma_0\}$ and $\varphi_1 = \{\mu_1, \sigma_1\}$.

In common applications of HMRFs, we observe only one instantiation $\mathbf{y}$ which is emitted according to the hidden state vector $\mathbf{x}$, and the task is to infer the most probable state configuration of $\mathbf{X}$, or to compute the marginal probabilities of $\mathbf{X}$. In both tasks, we need to estimate the parameters $\boldsymbol{\theta} = \{\theta_1, ..., \theta_r\}$ and $\boldsymbol{\varphi} = \{\varphi_0, ..., \varphi_{m-1}\}$. Usually, we seek *maximum likelihood estimates* of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ which maximize the log likelihood

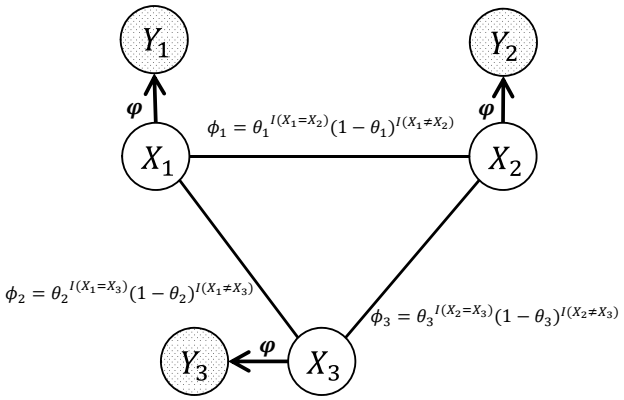

Figure 1: The pairwise HMRF model with three latent nodes (X_1, X_2, X_3) and observable nodes (Y_1, Y_2, Y_3) with parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$ and $\boldsymbol{\varphi} = \{\varphi_0, \varphi_1\}$.

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \log P(\mathbf{y};\boldsymbol{\theta},\boldsymbol{\varphi}) = \log \sum_{\mathbf{x}\in\mathbb{X}^\mathbf{d}} P(\mathbf{x},\mathbf{y};\boldsymbol{\theta},\boldsymbol{\varphi}). \quad (3)$$

Since we only have one instantiation $(\mathbf{x}, \mathbf{y})$, we usually have to assume that θ_i's are the same for $i = 1, ..., r$ for effective parameter learning. This *homogeneity* assumption is widely used in computer vision problems because people usually assume that the neighborhood system on an image is invariant across its different regions. Therefore, conventional HMRFs refer to homogeneous HMRFs, similar to conventional HMMs whose transition matrix is invariant across different sites.

2.2 Heterogeneous HMRFs

In a *heterogeneous* HMRF, the potential functions on different cliques can be different. Taking the model in Figure 1 as an example, θ_1, θ_2 and θ_3 can be different if the HMRF is heterogeneous. As with conventional HMRFs, we want to be able to address applications that have one instantiation $(\mathbf{x}, \mathbf{y})$ where $\mathbf{y}$ is observable and $\mathbf{x}$ is hidden. Therefore, learning an HMRF from one instantiation $\mathbf{y}$ is infeasible if we free all θ's. To partially free the parameters, we assume that there is some background knowledge $\mathbf{k} = \{k_1, ..., k_r\}$ about the parameters $\boldsymbol{\theta} = \{\theta_1, ..., \theta_r\}$ in the form of some unknown smooth mapping function which maps θ_i to k_i for $i = 1, ..., r$. The background knowledge describes how these potential functions are different across different cliques. Taking pairwise HMRFs for example, the potentials on the edges with similar background knowledge should have similar parameters. We can regard the homogeneity assumption in conventional HMRFs as an extreme type of background knowledge that $k_1 = k_2 = ... = k_r$. The problem we solve in this paper is *to estimate $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ which maximize the log likelihood $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ in Formula (3), subject to the condition that the estimate of $\boldsymbol{\theta}$ is smooth with respect to* $\mathbf{k}$.

3 Parameter Learning Methods

Learning heterogeneous HMRFs in above manner involves *three* difficulties, (i) the intractable $Z(\boldsymbol{\theta})$, (ii) the latent $\mathbf{x}$, and (iii) the heterogeneous $\boldsymbol{\theta}$. The way we handle the intractable $Z(\boldsymbol{\theta})$ is similar to using contrastive divergence [Hinton, 2002] to learn MRFs. We review contrastive divergence and its variations in Section 3.1. To handle the latent $\mathbf{x}$ in HMRF learning, we introduce an EM algorithm in Section 3.2, which is applicable to conventional HMRFs. In Section 3.3, we further address the heterogeneity of $\boldsymbol{\theta}$ in the M-step of the EM algorithm.

3.1 Contrastive Divergence for MRFs

Assume that we observe s independent samples $\mathfrak{X} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^s\}$ from (1), and we want to estimate $\boldsymbol{\theta}$. The log likelihood $\mathcal{L}(\boldsymbol{\theta}|\mathfrak{X})$ is concave w.r.t. $\boldsymbol{\theta}$, and we can use gradient ascent to find the MLE of $\boldsymbol{\theta}$. The partial derivative of $\mathcal{L}(\boldsymbol{\theta}|\mathfrak{X})$ with respect to θ_i is

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}|\mathfrak{X})}{\partial \theta_i} = \frac{1}{s}\sum_{j=1}^{s} \psi_i(\mathbf{x}^j) - E_{\boldsymbol{\theta}}\psi_i = E_{\mathfrak{X}}\psi_i - E_{\boldsymbol{\theta}}\psi_i, \quad (4)$$

where ψ_i is the sufficient statistic corresponding to θ_i, and $E_{\boldsymbol{\theta}}\psi_i$ is the expectation of ψ_i with respect to the distribution specified by $\boldsymbol{\theta}$. In the i-th iteration of gradient ascent, the parameter update is

$$\boldsymbol{\theta}_{(i+1)} = \boldsymbol{\theta}_{(i)} + \eta \nabla \mathcal{L}(\boldsymbol{\theta}_{(i)}|\mathfrak{X}) = \boldsymbol{\theta}_{(i)} + \eta(E_{\mathfrak{X}}\boldsymbol{\psi} - E_{\boldsymbol{\theta}_{(i)}}\boldsymbol{\psi}),$$

where η is the learning rate. However the exact computation of $E_{\boldsymbol{\theta}}\psi_i$ takes time exponential in the treewidth of $\mathbb{G}$. A few sampling-based methods have been proposed to solve this problem. The key differences among these methods are how to draw particles and how to compute $E_{\boldsymbol{\theta}}\boldsymbol{\psi}$ from the particles. MCMC-MLE [Geyer, 1991, Zhu and Liu, 2002] uses importance sampling, but might suffer from degeneracy when $\boldsymbol{\theta}_{(i)}$ is far away from $\boldsymbol{\theta}_{(1)}$. Contrastive divergence [Hinton, 2002] generates new particles in each iteration according to the current $\boldsymbol{\theta}_{(i)}$ and does not require the particles to reach equilibrium, so as to save computation. Variations of contrastive divergence include particle-filtered MCMC-MLE [Asuncion et al., 2010], persistent contrastive divergence (PCD) [Tieleman, 2008] and fast PCD [Tieleman and Hinton, 2009]. Because PCD is efficient and easy to implement, we employ it in this paper. Its pseudocode is provided in Algorithm 1. Other than contrastive divergence, MRF can be learned via ratio matching [Hyvärinen, 2007], non-local contrastive objectives [Vickrey et al., 2010], noise-contrastive estimation [Gutmann and Hyvärinen, 2010] and minimum KL contraction [Lyu, 2011].

Algorithm 1 PCD-n Algorithm [Tieleman, 2008]

Input: independent samples $\mathfrak{X} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^s\}$ from $P(\mathbf{x}; \boldsymbol{\theta})$, maximum iteration number T
Output: $\hat{\boldsymbol{\theta}}$ from the last iteration
Procedure:
Initialize $\boldsymbol{\theta}_{(1)}$ and initialize particles
Calculate $E_{\mathfrak{X}}\boldsymbol{\psi}$ from $\mathfrak{X}$
for $i = 1$ **to** T **do**
  Advance particles n steps under $\boldsymbol{\theta}_{(i)}$
  Calculate $E_{\boldsymbol{\theta}_{(i)}}\boldsymbol{\psi}$ from the particles
  $\boldsymbol{\theta}_{(i+1)} = \boldsymbol{\theta}_{(i)} + \eta(E_{\mathfrak{X}}\boldsymbol{\psi} - E_{\boldsymbol{\theta}_{(i)}}\boldsymbol{\psi})$
  Adjust η
end for

3.2 Expectation-Maximization for Learning Conventional HMRFs

We begin with a lower bound of the log likelihood function, and then introduce the EM algorithm which handles the latent variables in HMRFs. Let $q_{\mathbf{x}}(\mathbf{x})$ be any distribution on $\mathbf{x} \in \mathbb{X}^d$. It is well known that there exists a lower bound of the log likelihood $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ in (3), which is provided by an auxiliary function $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$ defined as follows,

$$\begin{aligned}\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}) &= \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}(\mathbf{x}) \log \frac{P(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\varphi})}{q_{\mathbf{x}}(\mathbf{x})} \\ &= \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) - \mathrm{KL}[q_{\mathbf{x}}(\mathbf{x}) | P(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varphi})],\end{aligned} \quad (5)$$

where $\mathrm{KL}[q_{\mathbf{x}}(\mathbf{x})|P(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varphi})]$ is the Kullback-Leibler divergence between $q_{\mathbf{x}}(\mathbf{x})$ and $P(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varphi})$, the posterior distribution of the hidden variables. This Kullback-Leibler divergence is the distance between $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$.

Expectation-Maximization: We maximize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ with an expectation-maximization (EM) algorithm which iteratively maximizes its lower bound $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$. We first initialize $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\varphi}^{(0)}$. In the t-th iteration, the updates in the expectation (E) step and the maximization (M) step are

$$q_{\mathbf{x}}^{(t)} = \arg\max_{q_{\mathbf{x}}} \mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\varphi}^{(t-1)}\}) \quad \text{(E)},$$

$$\boldsymbol{\theta}^{(t)}, \boldsymbol{\varphi}^{(t)} = \arg\max_{\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}} \mathcal{F}(q_{\mathbf{x}}^{(t)}, \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}) \quad \text{(M)}.$$

In the E-step, we maximize $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\varphi}^{(t-1)}\})$ with respect to $q_{\mathbf{x}}(\mathbf{x})$. Because the difference between $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$ and $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is $\mathrm{KL}[q_{\mathbf{x}}(\mathbf{x})|P(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\varphi})]$, the maximizer in the E-step $q_{\mathbf{x}}^{(t)}$ is $P(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\varphi}^{(t-1)})$, namely the posterior distribution of $\mathbf{x}|\mathbf{y}$ under the current estimated parameters $\boldsymbol{\theta}^{(t-1)}$ and $\boldsymbol{\varphi}^{(t-1)}$. This posterior distribution can be calculated by Markov chain Monte Carlo for general graphs.

In the M-step, we maximize $\mathcal{F}(q_{\mathbf{x}}^{(t)}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$ with respect to $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$, which can be rewritten as

$$\begin{aligned}
&\arg\max_{\{\boldsymbol{\theta},\boldsymbol{\varphi}\}} \mathcal{F}(q_{\mathbf{x}}^{(t)}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}) \\
=&\arg\max_{\{\boldsymbol{\theta},\boldsymbol{\varphi}\}} \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\varphi}) \\
=&\arg\max_{\{\boldsymbol{\theta},\boldsymbol{\varphi}\}} \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \Big\{ \log P(\mathbf{x}; \boldsymbol{\theta}) + \log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\varphi}) \Big\}.
\end{aligned}$$

It is obvious that this function can be maximized with respect to $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ separately as

$$\begin{aligned}
\boldsymbol{\theta}^{(t)} &= \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{x}; \boldsymbol{\theta}), \\
\boldsymbol{\varphi}^{(t)} &= \arg\max_{\boldsymbol{\varphi}} \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\varphi}).
\end{aligned} \tag{6}$$

Estimating $\boldsymbol{\varphi}$: Estimating $\boldsymbol{\varphi}$ in this maximum likelihood manner is straightforward, because the maximization can be rewritten as follows,

$$\begin{aligned}
&\arg\max_{\boldsymbol{\varphi}} \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\varphi}) \\
=&\arg\max_{\boldsymbol{\varphi}} \sum_{i=1}^{d} \sum_{x_i\in\mathbb{X}} q_{x_i}^{(t)}(x_i) \log P(y_i|x_i; \boldsymbol{\varphi}),
\end{aligned}$$

where $q_{\mathbf{x}}^{(t)}(\mathbf{x}) = \prod_{i=1}^{d} q_{x_i}^{(t)}(x_i)$.

Estimating $\boldsymbol{\theta}$: Estimating $\boldsymbol{\theta}$ in Formula (6) is difficult due to the intractable $Z(\boldsymbol{\theta})$. Some approaches [Zhang et al., 2001, Celeux et al., 2003] use pseudo-likelihood [Besag, 1975] to estimate $\boldsymbol{\theta}$ in the M-step. It can be shown that $\sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{x}; \boldsymbol{\theta})$ is *convex* with respect to $\boldsymbol{\theta}$. Therefore, we can use gradient ascent to find the MLE of $\boldsymbol{\theta}$, which is similar to using contrastive divergence [Hinton, 2002] to learn MRFs in Section 3.1.

Denote $\sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \log P(\mathbf{x}; \boldsymbol{\theta})$ by $\mathcal{L}_M(\boldsymbol{\theta}|q_{\mathbf{x}}^{(t)})$. The partial derivative of $\mathcal{L}_M(\boldsymbol{\theta}|q_{\mathbf{x}}^{(t)})$ with respect to θ_i is

$$\frac{\partial \mathcal{L}_M(\boldsymbol{\theta}|q_{\mathbf{x}}^{(t)})}{\partial \theta_i} = \sum_{\mathbf{x}\in\mathbb{X}^{\mathbf{d}}} q_{\mathbf{x}}^{(t)}(\mathbf{x}) \Big\{ \psi_i(\mathbf{x}) - E_{\boldsymbol{\theta}} \psi_i(\mathbf{x}) \Big\}.$$

Therefore, the derivative here is similar to the derivative in contrastive divergence in Formula (4) except we have to reweight it to $q_{\mathbf{x}}^{(t)}$. We run the EM algorithm until both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ converge. Note that when learning homogeneous HMRFs with this algorithm, we tie all θ's all the time, namely $\boldsymbol{\theta} = \{\theta\}$. Therefore, we name this parameter learning algorithm for conventional HMRFs the *EM-homo-PCD* algorithm.

3.3 Learning Heterogeneous HMRFs

Learning heterogeneous HMRFs is different from learning conventional homogeneous HMRFs in two ways. First, we need to free the θ's in heterogeneous HMRFs. Second, there is some background knowledge $\mathbf{k}$ about how the θ's are different, as introduced in Section 2. Therefore, we make two modifications to the EM-homo-PCD algorithm in order to learn heterogeneous HMRFs with background knowledge. First, we estimate the θ's separately, which obviously brings more variance in estimation. Second, within each iteration of the contrastive divergence update, we apply a kernel regression to smooth the estimate of the θ's with respect to the background knowledge $\mathbf{k}$. Specifically, in the i-th iteration of PCD update, we advance the particles under $\hat{\boldsymbol{\theta}}_{(i)}$ for n steps, and calculate the moments $E_{\hat{\boldsymbol{\theta}}_{(i)}} \boldsymbol{\psi}$ from the particles. Therefore, we can update the estimate as

$$\tilde{\boldsymbol{\theta}}_{(i+1)} = \hat{\boldsymbol{\theta}}_{(i)} + \eta \nabla \mathcal{L}_M(\boldsymbol{\theta}|q_{\mathbf{x}}^{(t)}).$$

Then we regress $\tilde{\boldsymbol{\theta}}_{(i+1)}$ with respect to $\mathbf{k}$ via Nadaraya-Watson kernel regression [Nadaraya, 1964, Watson, 1964], and set $\hat{\boldsymbol{\theta}}_{(i+1)}$ to be the fitted values. For ease of notation, we drop the iteration index $(i+1)$. Suppose that $\tilde{\boldsymbol{\theta}} = \{\tilde{\theta}_1, ..., \tilde{\theta}_r\}$ is the estimate before kernel smoothing; we set the smoothed estimate $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_1, ..., \hat{\theta}_r\}$ as

$$\hat{\theta}_j = \sum\nolimits_{i=1}^{r} \gamma_{ij} \tilde{\theta}_i, \forall j = 1, ..., r,$$

where

$$\gamma_{ij} = \frac{K(\frac{k_i - k_j}{h})}{\sum_{m=1}^{r} K(\frac{k_m - k_j}{h})}.$$

For the kernel function K, we use the Epanechnikov kernel, which is usually computationally more efficient than a Gaussian kernel. We tune the bandwidth h through cross-validation, namely we select the bandwidth which minimizes the leave-one-out score

$$\frac{1}{r} \sum\nolimits_{i=1}^{r} \left(\frac{\tilde{\theta}_i - \hat{\theta}_i}{1 - \gamma_{ii}} \right)^2.$$

Tuning the bandwidth is usually computation-intensive, so we tune it every t_0 iterations to save computation. We name our parameter learning algorithm for heterogeneous HMRFs the *EM-kernel-PCD* algorithm. Its pseudo-code is given in Algorithm 2.

Another intuitive way of handling background knowledge about these heterogeneous parameters is to create bins according to the background knowledge and tie the θ's that are in the same bin. Suppose that we have b bins after we carefully select the binwidth,

Algorithm 2 EM-kernel-PCD Algorithm

Input: sample $\mathbf{y}$, background knowledge $\mathbf{k}$, max iteration number T, initial bandwidth h
Output: $\hat{\boldsymbol{\theta}}$ from the last iteration
Procedure:
Initialize $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{\varphi}}$ and particles
while not converge **do**
  E-step: infer $\hat{\mathbf{x}}$ from $\mathbf{y}$
  Calculate $E_{\hat{\mathbf{x}}}\boldsymbol{\psi}$ from $\hat{\mathbf{x}}$
  for $i = 1$ **to** T **do**
    Advance particles for n steps under $\hat{\boldsymbol{\theta}}_{(i)}$
    Calculate $E_{\hat{\boldsymbol{\theta}}_{(i)}}\boldsymbol{\psi}$ from the particles
    $\tilde{\boldsymbol{\theta}}_{(i+1)} = \hat{\boldsymbol{\theta}}_{(i)} + \eta \nabla \mathcal{L}_M(\boldsymbol{\theta}|q_{\mathbf{x}}^{(t)})$
    $\hat{\boldsymbol{\theta}}_{(i+1)} = kernelRegFit(\tilde{\boldsymbol{\theta}}_{(i+1)}, \mathbf{k}, h)$
    Adjust η and tune bandwidth h
  end for
  MLE $\hat{\boldsymbol{\varphi}}$ from $\hat{\mathbf{x}}$ and $\mathbf{y}$
end while

namely we have $\boldsymbol{\theta} = \{\theta_1, ..., \theta_b\}$. The rest of the algorithm is the same as the EM-homo-PCD algorithm in Section 3.2. We name this parameter learning algorithm via binning the *EM-binning-PCD* algorithm. We can also regard our EM-kernel-PCD algorithm as a soft-binning version of EM-binning-PCD.

4 Simulations

We investigate the performance of our EM-kernel-PCD algorithm on heterogeneous HMRFs with different structures, namely a tree-structure HMRF and a grid-structure HMRF. In the simulations, we first set the ground truth of the parameters, and then set the background knowledge. We then generate one example $\mathbf{x}$ and then generate one example $\mathbf{y}|\mathbf{x}$. With the observable $\mathbf{y}$, we apply EM-kernel-PCD, EM-binning-PCD and EM-homo-PCD to learn the parameters $\boldsymbol{\theta}$. We eventually compare the three algorithms by their average absolute estimate error $1/r \sum_{i=1}^{r} |\theta_i - \hat{\theta}_i|$ where $\hat{\theta}_i$ is the estimate of θ_i.

For the HMRFs, each dimension of $\mathbf{X}$ takes values in $\{0, 1\}$. The pairwise potential function ϕ_i on edge i (connecting X_u and X_v) parameterized by θ_i ($0 < \theta_i < 1$) is $\phi_i(\mathbf{X}; \theta_i) = \theta_i^{I(X_u = X_v)}(1 - \theta_i)^{I(X_u \neq X_v)}$ for $i = 1, 2, 3$, where I is an indicator variable. For the tree structure, we choose a perfect binary tree of height 12, which yields a total number of 8,191 nodes and 8,190 parameters, i.e. $d = 8{,}191$ and $r = 8{,}190$. For the grid-structure HMRFs, we choose a grid of 100 rows and 100 columns, which yields a total number of 10,000 nodes and 19,800 parameters, i.e. $d = 10{,}000$ and $r = 19{,}800$. For both of the two models, we generate $\theta_i \sim U(0.5, 1)$ independently and then generate the background knowledge k_i. We have two types of

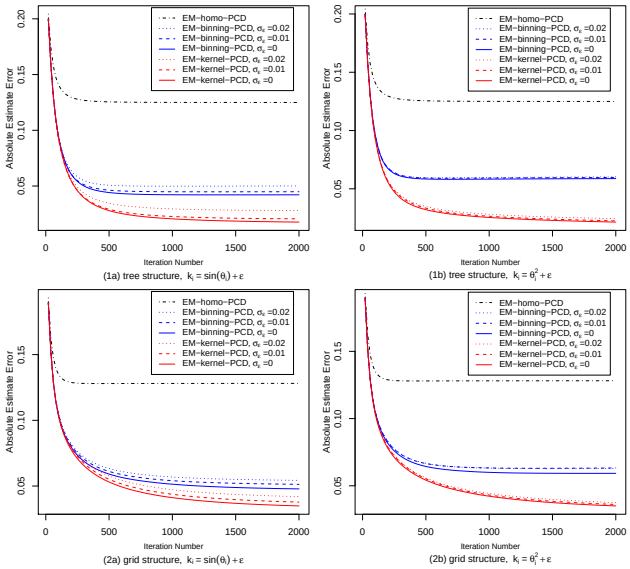

Figure 2: Performance of EM-homo-PCD, EM-binning-PCD and EM-kernel-PCD in tree-HMRFs and grid-HMRFs for two types of background knowledge: (a) $k_i = \sin\theta_i + \epsilon$, and (b) $k_i = \theta_i^2 + \epsilon$.

background knowledge. In the first type of background knowledge, we set $k_i = \sin\theta_i + \epsilon$. In the second type of background knowledge, we set $k_i = \theta_i^2 + \epsilon$, where ϵ is some random Gaussian noise from $N(0, \sigma_\epsilon^2)$. We try three values for σ_ϵ, namely 0.0, 0.01 and 0.02. Then we generate one instantiation $\mathbf{x}$. Finally, we generate one observable $\mathbf{y}$ from a d dimensional multivariate normal distribution $N(\mu\mathbf{x}, \sigma^2\mathbb{I})$ where $\mu = 2$ is the strength of signal, and $\sigma^2 = 1.0$ is the variance of the manifestation, and $\mathbb{I}$ is the identity matrix of dimension d. For our EM-kernel-PCD algorithm, we use an Epanechnikov kernel with $\alpha = \beta = 5$. For tuning bandwidth h, we try 100 values in total, namely 0.005, 0.01, 0.015, ..., 0.5. For the EM-binning-PCD algorithm, we set the binwidth to be 0.005. The rest of the parameter settings for the three algorithms are the same, including the n parameter in PCD which is set to be 1 and the number of particles which is set to be 100. We also replicate each experiment 20 times, and the averaged results are reported.

Performance of the algorithms The results from the tree-structure HMRFs and the grid-structure HMRFs are reported in Figure 2. We plot the average absolute error of the estimate of the three algorithms against the number of iterations of PCD update. We have separate plots for background knowledge $k_i = \sin\theta_i + \epsilon$, and background knowledge $k_i = \theta_i^2 + \epsilon$. Since there are three noise levels for background knowledge, both the EM-kernel-PCD algorithm and the EM-binning-PCD algorithm have three variations. All the three algorithms converge as they iterate. It

is observed that the absolute estimate error of the EM-homo-PCD algorithm reduces to 0.125 as it converges. Since the parameters θ_i's are drawn independently from the uniform distribution on the interval $[0.5, 1]$, the EM-homo-PCD algorithm ties all the θ_i's and estimates them to be 0.75. Therefore, the averaged absolute error is $\int_{0.5}^{1.0} 2|x - 0.75| \, \mathrm{d}x = 0.125$. Our EM-kernel-PCD algorithm significantly outperforms the EM-binning-PCD algorithm and the EM-homo-PCD algorithm. It is also observed that as the noise level of background knowledge increases, the performance of the EM-kernel-PCD algorithm and the EM-binning-PCD algorithm deteriorates. However, as long as the noise level is moderate, the performance of our EM-kernel-PCD algorithm is satisfactory. The results from the tree-structure HMRFs and the grid-structure HMRFs are comparable except that it takes more iterations to converge in grid-structure HMRFs than in tree-structure HMRFs.

Behavior of the algorithms We then plot the estimated parameters against their background knowledge in the iterations of our EM-kernel-PCD algorithm. We provide plots for after 100 iterations, after 200 iterations and after convergence respectively, to show how the EM-kernel-PCD algorithm behaves during the gradient ascent. Figure 3 shows the plots for the background knowledge $k_i = \sin\theta_i + \epsilon$ and the background knowledge $k_i = \theta_i^2 + \epsilon$ with three levels of noise (namely σ_ϵ=0, 0.01 and 0.02) for both the tree-structure HMRFs and the grid-structure HMRFs. It is observed that as the algorithm iterates, it gradually recovers the relationship between the parameters and the background knowledge. There is still a gap between our estimate and the ground truth. This is because we only have one hidden instantiation $\mathbf{x}$ and we have to infer $\mathbf{x}$ from the observed $\mathbf{y}$ in the E-step. Especially at the boundaries, we can observe a certain amount of estimate bias. The boundary bias is very common in kernel regression problems because there are fewer data points at the boundaries [Fan, 1992].

Choosing parameter n One parameter in contrastive divergence algorithms is n, the number of MCMC steps we need to perform under the current parameters in order to generate the particles. The rationale of contrastive divergence is that it is enough to find the direction to update the parameters by a few MCMC steps using the current parameters, and we do not have to reach the equilibrium. Therefore, the parameter n is usually set to be very small to save computation when we are learning general Markov random fields. Here we explore how we should choose the n parameter in our EM-kernel-PCD algorithm for learning HMRFs. We choose three values for n in the simulations, namely 1, 5 and 10. In Figure 4, the running time and absolute estimate error are plotted for the three choices in the tree-structure HMRFs and grid-structure HMRFs under different levels of noise in the background knowledge $k_i = \sin\theta_i + \epsilon$ and the background knowledge $k_i = \theta_i^2 + \epsilon$. The running time increases as n increases, but the estimation accuracy does not increase. This observation stays the same for different structures and different levels of noise in different types of background knowledge. This suggests that we can simply choose $n = 1$ in our EM-kernel-PCD algorithm.

5 Real-world Application

We use our EM-kernel-PCD algorithm to learn a heterogeneous HMRF model in a real-world genome-wide association study on breast cancer. The dataset is from NCI's Cancer Genetics Markers of Susceptibility (CGEMS) study [Hunter et al., 2007]. In total, 528,173 genetic markers (single-nucleotide polymorphisms or SNPs) for 1,145 breast cancer cases and 1,142 controls are genotyped on the Illumina HumanHap500 array, and the task is to identify the SNPs which are associated with breast cancer. This dataset has been used in the study of Liu et al. [2012b]. We build a heterogeneous HMRF model to identify the associated SNPs. In the HMRF model, the hidden vector $\mathbf{X} \in \{0, 1\}^d$ denotes whether the SNPs are associated with breast cancer, i.e. $X_i = 1$ means that the SNP_i is associated with breast cancer. For each SNP, we can perform a two-proportion z-test from the minor allele count in cases and the minor allele count in controls. Denote Y_i to be the test statistic from the two-proportion z-test for SNP_i. It can be derived that $Y_i|X_i{=}0 \sim N(0, 1)$ and $Y_i|X_i{=}1 \sim N(\mu_1, 1)$ for some unknown μ_1 ($\mu_1 \neq 0$). We assume that $\mathbf{X}$ forms a pairwise Markov random field with respect to the graph $\mathbb{G}$. The graph $\mathbb{G}$ is built as follows. We query the squared correlation coefficients (r^2 values) among the SNPs from HapMap [The International HapMap Consortium, 2003]. Each SNP becomes a node in the graph. For each SNP, we connect it with the SNP having the highest r^2 value with it. We also remove the edges whose r^2 values are below 0.25. There are in total 340,601 edges in the graph. The pairwise potential function ϕ_i on edge i (connecting X_u and X_v) parameterized by θ_i ($0 < \theta_i < 1$) is $\phi_i(\mathbf{X}; \theta_i) = \theta_i^{I(X_u = X_v)}(1 - \theta_i)^{I(X_u \neq X_v)}$ for $i = 1, ..., 340{,}601$, where I is an indicator variable. It is believed that two SNPs with a higher level of correlation are more likely to agree in their association with breast cancer. Therefore, we set the background knowledge $\mathbf{k}$ about the parameters to be the r^2 values between the SNPs on the edge. We first perform the two-proportion z-test and set $\mathbf{y}$ to be the calculated test statistics. Then we estimate $\boldsymbol{\theta}|\mathbf{y}, \mathbf{k}$ in the heterogeneous HMRF with respect to $\mathbb{G}$ using our EM-

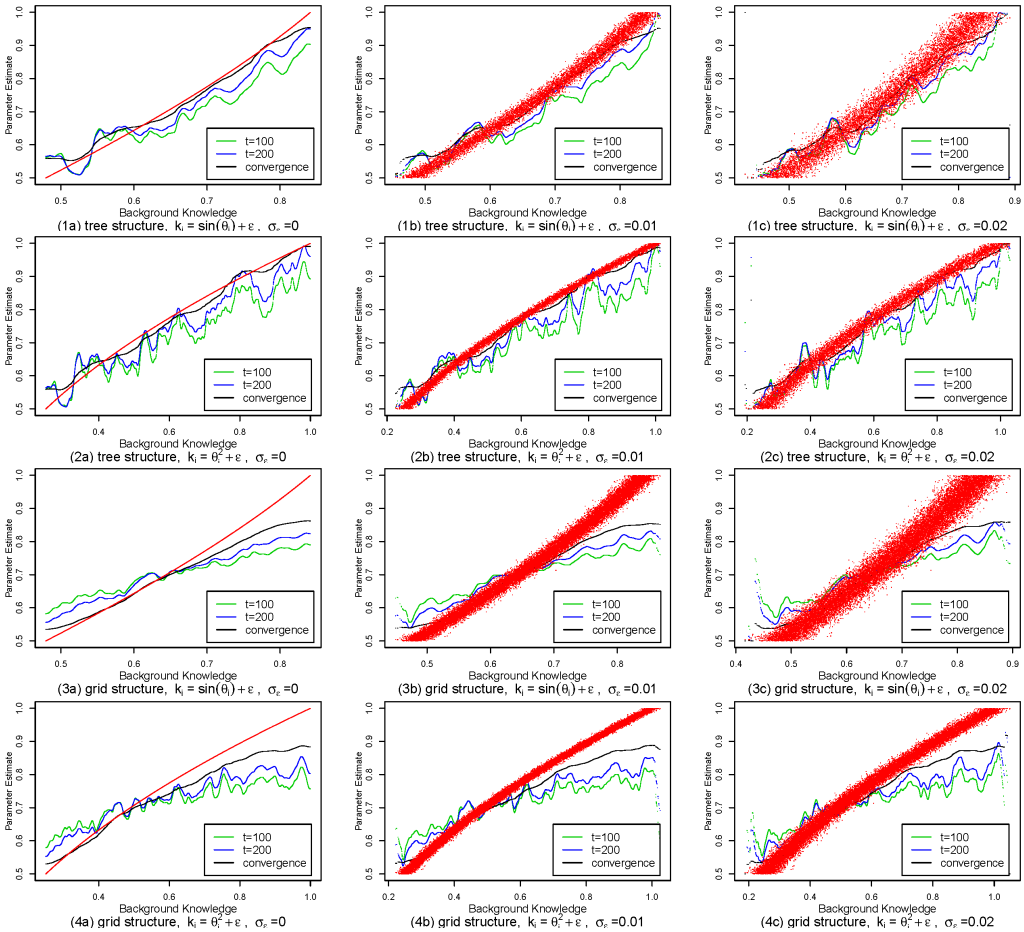

Figure 3: The behavior of the EM-kernel-PCD algorithm during gradient ascent for different types of background knowledge with different levels of noise in the tree-structure HMRFs and the grid-structure HMRFs. The red dots show the mapping pattern between the ground truth of the parameters and their background knowledge.

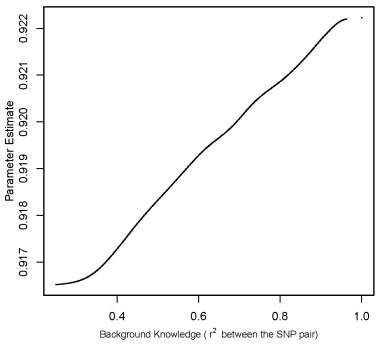

Figure 5: The estimated parameters against their background knowledge, namely the r^2 values between the pairs of SNPs.

kernel-PCD algorithm. After we estimate $\boldsymbol{\theta}$ and μ_1, we calculate the marginal probabilities of the hidden $\mathbf{X}$. Eventually, we rank the SNPs by the marginal probabilities $P(X_i = 1|\mathbf{y}; \hat{\boldsymbol{\theta}}, \hat{\mu}_1)$, and select the SNPs with the largest marginal probabilities.

The algorithm ran for 46 days on a single processor (AMD Opteron Processor, 3300 MHz) before it converged. We plotted the estimated parameters against their background knowledge, namely the r^2 values between the pairs of SNPs on the edges. The plot is provided in Figure 5. It is observed that the mapping between the estimated parameters and the background knowledge is monotone increasing, as we expect. Finally we calculated the marginal probabilities of the hidden $\mathbf{X}$, and ranked the SNPs by the marginal probabilities $P(X_i = 1|\mathbf{y}; \hat{\boldsymbol{\theta}}, \hat{\mu}_1)$. There are in total five SNPs with $P(X_i = 1|\mathbf{y}; \hat{\boldsymbol{\theta}}, \hat{\mu}_1)$ greater than 0.99, which means they are associated with breast cancer with a probability greater than 0.99 given the observed test statistics $\mathbf{y}$ under the estimated parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\mu}_1$. There is strong evidence in the literature that supports the association with breast cancer for three of them. The two SNPs rs2420946 and rs1219648 on chromosome 10 are reported by Hunter et al (2007), and have been further validated by 1,776 cases and 2,072 controls from three additional studies. Their associated gene FGFR2 is very well known to be associated with breast cancer in the literature. There is also strong evidence supporting the association of the SNP rs7712949

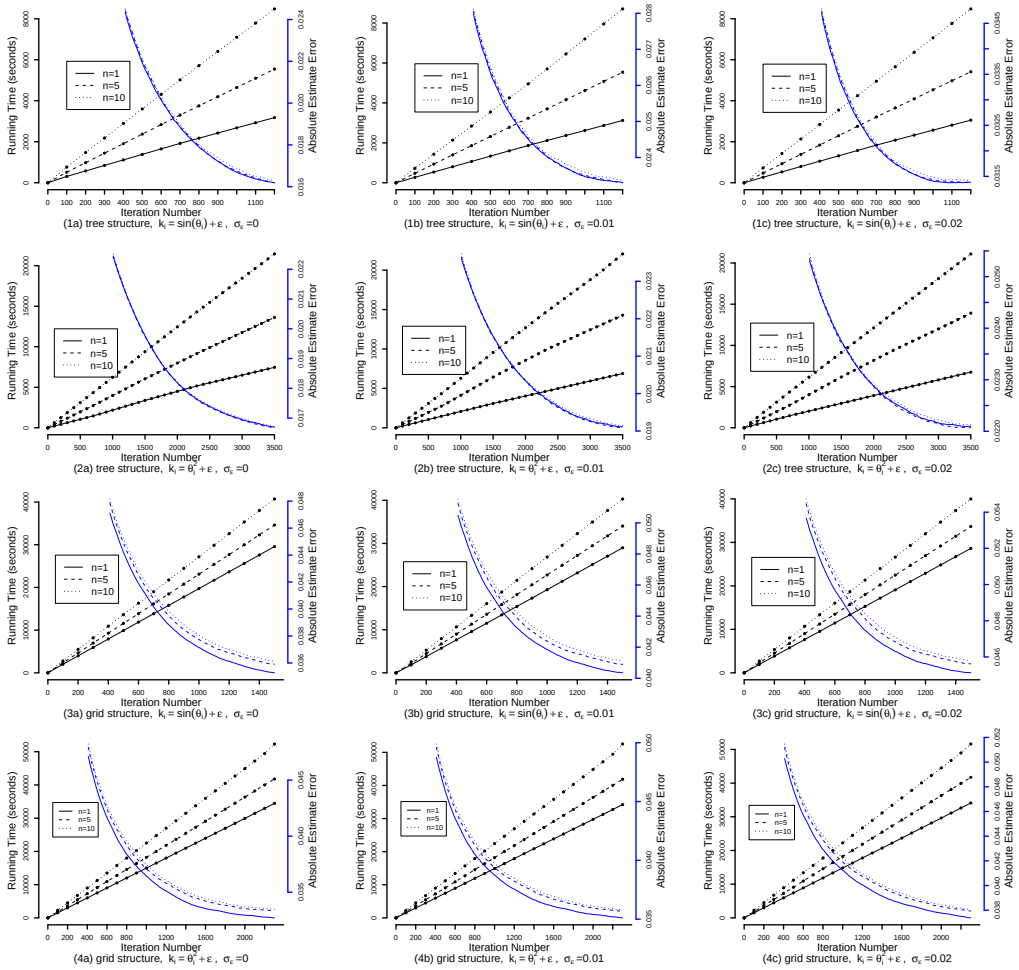

Figure 4: Absolute estimate error (plotted in blue, in the units on the right axes) and running time (plotted in black, in minutes on the left axes) of the EM-kernel-PCD algorithm in the tree-structure HMRFs and the grid-structure HMRFs when we choose different n values; n is the number of MCMC steps for advancing particles in the PCD algorithm. The absolute estimate error in the first 400 iterations is not shown in the plots.

on chromosome 5. The SNP rs7712949 is highly correlated (r^2=0.948) with SNP rs4415084 which has been identified to be associated with breast cancer by another six large-scale studies. [1]

6 Conclusion

Capturing parameter heterogeneity is an important issue in machine learning and statistics, and it is particular challenging in HMRFs due to both the intractable $Z(\boldsymbol{\theta})$ and the latent $\mathbf{x}$. In this paper, we propose the EM-kernel-PCD algorithm for learning the heterogeneous parameters with background knowledge. Our algorithm is built upon the PCD algorithm which handles the intractable $Z(\boldsymbol{\theta})$. The EM part we add is for dealing with the hidden $\mathbf{x}$. The kernel smoothing part we add is to adaptively incorporate the background knowledge about the heterogeneity in parameters in the gradient ascent learning. Eventually, the relation between the parameters and the background knowledge is recovered in a nonparametric way, which is also adaptive to the data. Simulations show that our algorithm is effective for learning heterogeneous HMRFs and outperforms alternative binning methods.

Similar to other EM algorithms, our algorithm only converges to a local maximum of the likelihood $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$, although the lower bound $\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}), \{\boldsymbol{\theta}, \boldsymbol{\varphi}\})$ nondecreases over the EM iterations (except for some MCMC error introduced in the E-step). Our algorithm also suffers from long run time due to computationally expensive PCD algorithm within each M-step. These two issues are important directions for future work.

Acknowledgements

The authors gratefully acknowledge the support of NIH grants R01GM097618, R01CA165229, R01LM010921, P30CA014520, UL1TR000427, NSF grants DMS-1106586, DMS-1308872 and the UW Carbone Cancer Center.

[1] http://snpedia.com/index.php/rs4415084

References

A. U. Asuncion, Q. Liu, A. T. Ihler, and P. Smyth. Particle filtered MCMC-MLE with connections to contrastive divergence. In *ICML*, 2010.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *NIPS*, 2002.

J. Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195, 1975.

G. Celeux, F. Forbes, and N. Peyrard. EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pattern Recognition*, 36:131–144, 2003.

S. P. Chatzis and G. Tsechpenakis. The infinite hidden Markov random field model. In *ICCV*, 2009.

S. P. Chatzis and T. A. Varvarigou. A fuzzy clustering approach toward hidden Markov random field models for enhanced spatially constrained image segmentation. *IEEE Transactions on Fuzzy Systems*, 16: 1351 – 1361, 2008.

J. Fan. Design-adaptive nonparametric regression. *Journal of the American Statistical Association*, 87 (420):998–1004, 1992. ISSN 01621459.

J. V. Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *ICML*, 2008.

C. J. Geyer. Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics*, pages 156–163, 1991.

Z. Ghahramani and M. I. Jordan. Factorial Hidden markov models. *Machine Learning*, 29:245–273, 1997.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14: 1771–1800, 2002.

D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *COLT*, 2009.

D. J. Hunter, P. Kraft, K. B. Jacobs, D. G. Cox, M. Yeager, S. E. Hankinson, S. Wacholder, Z. Wang, R. Welch, A. Hutchinson, J. Wang, K. Yu, N. Chatterjee, N. Orr, W. C. Willett, G. A. Colditz, R. G. Ziegler, C. D. Berg, S. S. Buys, C. A. Mccarty, H. S. Feigelson, E. E. Calle, M. J. Thun, R. B. Hayes, M. Tucker, D. S. Gerhard, J. F. Fraumeni, R. N. Hoover, G. Thomas, and S. J. Chanock. A genome-wide association study identifies alleles in fgfr2 associated with risk of sporadic postmenopausal breast cancer. *Nature Genetics*, 39(7):870–874, 2007.

A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5): 2499–2512, 2007.

J. Kim and R. Zabih. Factorial Markov random fields. In *ECCV*, pages 321–334, 2002.

X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV*, pages 269–282, 2006.

J. Liu, C. Zhang, C. McCarty, P. Peissig, E. Burnside, and D. Page. High-dimensional structured feature screening using binary Markov random fields. In *AISTATS*, 2012a.

J. Liu, C. Zhang, C. McCarty, P. Peissig, E. Burnside, and D. Page. Graphical-model based multiple testing under dependence, with applications to genome-wide association studies. In *UAI*, 2012b.

S. Lyu. Unifying non-maximum likelihood learning objectives with minimum KL contraction. In *NIPS*, pages 64–72, 2011.

J. C. Marioni, N. P. Thorne, and S. Tavaré. BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data. *Bioinformatics*, 22:1144–1146, 2006.

E. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9(1):141–142, 1964.

L. Song, B. Boots, S. Siddiqi, G. Gordon, and A. Smola. Hilbert space embeddings of hidden Markov models. In *ICML*, 2010.

The International HapMap Consortium. The international hapmap project. *Nature*, 426:789–796, 2003.

T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML*, 2008.

T. Tieleman and G. Hinton. Using fast weights to improve persistent contrastive divergence. In *ICML*, 2009.

D. Vickrey, C. Lin, and D. Koller. Non-local contrastive objectives. In *ICML*, 2010.

G. S. Watson. Smooth regression analysis. *The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964.

Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 2001.

S. C. Zhu and X. Liu. Learning in Gibbsian fields: How accurate and how fast can it be? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24: 1001–1006, 2002.