Define the marginal probability that \( H_i \) is null given all the observed statistics \( x \) under the parameters in \( \vartheta \), \( P_{\vartheta}(\theta_i = 0|x) \), to be the local index of significance (LIS) for \( H_i \) (Sun & Cai, 2009). If we can accurately calculate the posterior marginal probabilities of \( \theta \) (or LIS), then we can use a step-up procedure to control FDR at the nominal level \( \alpha \) as follows (Sun & Cai, 2009). We first sort LIS from the smallest value to the largest value. Suppose LIS\(_1, \) LIS\(_2, \) ..., and LIS\(_m\) are the ordered LIS, and the corresponding hypotheses are \( H_{(1)}, H_{(2)}, \ldots, \) and \( H_{(m)} \). Let

\[
k = \max \left\{ i : \frac{1}{i} \sum_{j=1}^{i} \text{LIS}(j) \leq \alpha \right\}.
\]

Then we reject \( H_{(i)} \) for \( i = 1, \ldots, k \).

Therefore, the key inferential problem that we need to solve is that of computing the posterior marginal distribution of the hidden variables \( \theta \), given the test statistics \( x \), namely \( P_{\vartheta}(\theta_i = 0|x) \), for \( i = 1, \ldots, m \). It is a typical inference problem if the parameters in \( \vartheta \) are known. Section 2.3 provides possible inference algorithms for calculating \( P_{\vartheta}(\theta_i = 0|x) \) for given \( \vartheta \). However, \( \vartheta \) is usually unknown in real-world applications, and we need to estimate it. Section 2.4 provides a novel EM algorithm for parameter learning in our MRF-coupled mixture model.

### 2.3 Posterior Inference

Now we are interested in calculating \( P_{\vartheta}(\theta_i = 0|x) \) for a given parameter set \( \vartheta \). One popular family of inference algorithms is the sum-product family (Kschischang et al., 2001), also known as belief propagation (Yedidia et al., 2000). For loop-free graphs, belief propagation algorithms provide exact inference results with a computational cost linear in the number of variables. In our MRF-coupled mixture model, the structure of the latent MRF is described by a graph \( G(V,E) \). When \( G \) is chain structured, the instantiation of belief propagation is the forward-backward algorithm (Baum et al., 1970). When \( G \) is tree structured, the instantiation of belief propagation is the upward-downward algorithm (Crouse et al., 1998). For graphical models with cycles, loopy belief propagation (Murphy et al., 1999; Weiss, 2000) and the tree-reweighted algorithm (Wainwright et al., 2003a) can be used for approximate inference. Other inference algorithms for graphical models include junction trees (Lauritzen & Spiegelhalter, 1988), sampling methods (Gelfand & Smith, 1990), and variational methods (Jordan et al., 1999). Recent papers (Schraudolph & Kamenetsky, 2009; Schraudolph, 2010) discuss exact inference algorithms on binary Markov random fields which allow loops. In our simulations, we use belief propagation when the graph \( G \) has no loops. When \( G \) has loops (e.g., in the simulations on genetic data and the real-world application), we use a Markov chain Monte Carlo (MCMC) algorithm to perform inference for \( P_{\vartheta}(\theta_i = 0|x) \).

### 2.4 Parameters and Parameter Learning

In our procedure, the dependence among these hypotheses is represented by a graphical model on the latent vector \( \theta \) parameterized by \( \phi \), and observed test statistics \( x \) are represented by the coupled mixture parameterized by \( \psi \). In Sun and Cai’s work on HMMs, \( \phi \) is the transition parameter and \( \psi \) is the emission parameter. One implicit assumption in their work is that the transition parameter and the emission parameter stay the same for \( i(i = 1, \ldots, m) \). Our extension to MRFs also allows us to untie these parameters. In the second set of basic simulations in Section 3, we make \( \phi \) and \( \psi \) heterogeneous and investigate how this affects the numerical performance. In the simulations on genetic data in Section 4 and the real-world GWAS application in Section 5, we have different parameters for SNP pairs with different levels of correlation.

In our model, learning \( (\phi, \psi) \) is difficult for two reasons. First, learning parameters is difficult by nature in undirected graphical models due to the global normalization constant (Wainwright et al., 2003b; Welling & Sutton, 2005). State-of-the-art MRF parameter learning methods include MCMC-MLE (Geyer, 1991), contrastive divergence (Hinton, 2002) and variational methods (Ganapathi et al., 2008). Several new sampling methods with higher efficiency have been recently proposed, such as persistent contrastive divergence (Tieleman, 2008), fast-weight contrastive divergence (Tieleman & Hinton, 2009), tempered transitions (Salakhutdinov, 2009), and particle-filtered MCMC-MLE (Asuncion et al., 2010). In our procedure, we use the persistent contrastive divergence algorithm to estimate parameters \( \phi \). Another difficulty is that \( \theta \) is latent and we only have one observed training sample \( x \). We use an EM algorithm to solve this problem. In the E-step, we run our MCMC algorithm in Section 2.3 to infer the latent \( \theta \) based on the currently estimated parameters \( \vartheta = (\phi, \psi) \). In the M-step, we run the persistent contrastive divergence (PCD) algorithm (Tieleman, 2008) to estimate \( \phi \) from the currently inferred \( \theta \). Note that PCD is also an iterative algorithm, and we run it until it converges in each M-step. In the M-step, we also do a maximum likelihood estimation of \( \psi \) from the currently inferred \( \theta \) and observed \( x \). We run the EM algorithm until both \( \phi \) and \( \psi \) converge. Although this EM algorithm involves intensive computation in both E-step and M-step, it converges very quickly in our experiments.