Chapter 28

Nonparametric Graphical Models

In this chapter we discuss some nonparametric methods for graphical modeling. In the discrete case, where the data are binary or drawn from a finite alphabet, Markov random fields are already essentially nonparametric, since the cliques can take only a finite number of values. Continuous data are different. The Gaussian graphical model is the standard parametric model for continuous data, but it makes distributional assumptions that are often unrealistic. We discuss two approaches to building more flexible graphical models. One allows arbitrary graphs and a nonparametric extension of the Gaussian; the other uses kernel density estimation and restricts the graphs to trees and forests.

28.1 Introduction

This chapter presents two methods for constructing nonparametric graphical models for continuous data. In the discrete case, where the data are binary or drawn from a finite alphabet, Markov random fields or log-linear models are already essentially nonparametric, since the cliques can take only a finite number of values. Continuous data are different. The Gaussian graphical model is the standard parametric model for continuous data, but it makes distributional assumptions that are typically unrealistic. Yet few practical alternatives to the Gaussian graphical model exist, particularly for high dimensional data. We discuss two approaches to building more flexible graphical models that exploit sparsity. These two approaches are at different extremes in the array of choices available. One allows arbitrary graphs, but makes a distributional restriction through the use of copulas; this is a semiparametric extension of the Gaussian. The other approach uses kernel density estimation and restricts the graphs to trees and forests; in this case the model is fully nonparametric, at the expense of structural restrictions. We describe two-step estimation methods for both approaches. We also outline some statistical theory for the methods, and compare them in some examples. The primary references for this material are Liu et al. (2009) and Liu et al.

(2011).

The methods we present here are relatively simple, and many more possibilities remain for nonparametric graphical modeling. But one of the main messages of this chapter is that a little nonparametricity can go a long way.

28.2 Two Families of Nonparametric Graphical Models

The graph of a random vector is a useful way of exploring the underlying distribution. Recall that if $X = (X_1, \ldots, X_d)$ is a random vector with distribution P, then the undirected graph $G = (V, E)$ corresponding to P consists of a vertex set V and an edge set E where V has d elements, one for each variable X_i. The edge between (i, j) is excluded from E if and only if X_i is independent of X_j given the other variables $X_{\backslash\{i,j\}} \equiv (X_s : 1 \leq s \leq d, \;\; s \neq i, j)$, written

$$X_i \perp\!\!\!\perp X_j \;\Big|\; X_{\backslash\{i,j\}}. \tag{28.1}$$

The general form for a (strictly positive) probability density encoded by an undirected graph G is

$$p(x) = \frac{1}{Z(f)} \exp\left(\sum_{C \in \text{Cliques}(G)} f_C(x_C) \right), \tag{28.2}$$

where the sum is over all cliques, or fully connected subsets of vertices of the graph. In general, this is what we mean by a *nonparametric graphical model*. It is the graphical model analogue of the general nonparametric regression model. Model (28.2) has two main ingredients, the graph G and the functions $\{f_C\}$. However, without further assumptions, it is much too general to be practical. The main difficulty in working with such a model is the normalizing constant $Z(f)$, which cannot, in general, be efficiently computed or approximated.

In the spirit of nonparametric estimation, we can seek to impose structure on either the graph or the functions f_C in order to get a flexible and useful family of models. One approach parallels the ideas behind sparse additive models for regression. Specifically, we replace the random variable $X = (X_1, \ldots, X_d)$ by the transformed random variable $f(X) = (f_1(X_1), \ldots, f_d(X_d))$, and assume that $f(X)$ is multivariate Gaussian. This results in a nonparametric extension of the Normal that we call the *nonparanormal* distribution. The nonparanormal depends on the univariate functions $\{f_j\}$, and a mean μ and covariance matrix Σ, all of which are to be estimated from data. While the resulting family of distributions is much richer than the standard parametric Normal (the paranormal), the independence relations among the variables are still encoded in the precision matrix $\Omega = \Sigma^{-1}$, as we show below.

The second approach is to force the graphical structure to be a tree or forest, where each pair of vertices is connected by at most one path. Thus, we relax the distributional assumption of normality but we restrict the allowed family of undirected graphs. The complexity of the model is then regulated by selecting the edges to include, using cross validation.

	nonparanormal	*forest densities*
univariate marginals	nonparametric	nonparametric
bivariate marginals	determined by Gaussian copula	nonparametric
graph	unrestricted	acyclic

Figure 28.1. *Comparison of properties of the nonparanormal and forest-structured densities.*

Figure 28.1 summarizes the tradeoffs made by these two families of models. The nonparanormal can be thought of as an extension of additive models for regression to graphical modeling. This requires estimating the univariate marginals; in the copula approach, this is done by estimating the functions $f_j(x) = \mu_j + \sigma_j \Phi^{-1}(F_j(x))$, where F_j is the distribution function for variable X_j. After estimating each f_j, we transform to (assumed) jointly Normal via $Z = (f_1(X_1), \ldots, f_d(X_d))$ and then apply methods for Gaussian graphical models to estimate the graph. In this approach, the univariate marginals are fully nonparametric, and the sparsity of the model is regulated through the inverse covariance matrix, as for the graphical lasso, or "glasso" (Banerjee et al., 2008; Friedman et al., 2007)[27] The model is estimated in a two-stage procedure; first the functions f_j are estimated, and then inverse covariance matrix Ω is estimated. The high level relationship between linear regression models, Gaussian graphical models, and their extensions to additive and high dimensional models is summarized in Figure 28.2.

In the forest graph approach, we restrict the graph to be acyclic, and estimate the bivariate marginals $p(x_i, x_j)$ nonparametrically. In light of equation (28.27), this yields the full nonparametric family of graphical models having acyclic graphs. Here again, the estimation procedure is two-stage; first the marginals are estimated, and then the graph is estimated. Sparsity is regulated through the edges (i, j) that are included in the forest.

Clearly these are just two tractable families within the very large space of possible nonparametric graphical models specified by equation (28.2). Many interesting research possibilities remain for novel nonparametric graphical models that make different assumptions; we discuss some possibilities in a concluding section. We now discuss details of these two model families, beginning with the nonparanormal.

[27] Throughout the chapter we use the term graphical lasso, or glasso, coined by Friedman et al. (2007) to refer to the solution obtained by ℓ_1-regularized log-likelihood under the Gaussian graphical model. This estimator goes back at least to Yuan and Lin (2007), and an iterative lasso algorithm for doing the optimization was first proposed by Banerjee et al. (2008). In our experiments we use the R packages `glasso` (Friedman et al., 2007) and `huge` to implement this algorithm.

assumptions	*dimension*	*regression*	*graphical Models*
parametric	low	linear model	multivariate Normal
	high	lasso	graphical lasso
nonparametric	low	additive model	nonparanormal
	high	sparse additive model	sparse nonparanormal

Figure 28.2. *Comparison of regression and graphical models. The nonparanormal extends additive models to the graphical model setting. Regularizing the inverse covariance leads to an extension to high dimensions, which parallels sparse additive models for regression.*

28.3 The Nonparanormal

We say that a random vector $X = (X_1, \ldots, X_d)^T$ has a *nonparanormal* distribution and write

$$X \sim NPN(\mu, \Sigma, f)$$

in case there exist functions $\{f_j\}_{j=1}^d$ such that $Z \equiv f(X) \sim N(\mu, \Sigma)$, where $f(X) = (f_1(X_1), \ldots, f_d(X_d))$. When the f_j's are monotone and differentiable, the joint probability density function of X is given by

$$p_X(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}\left(f(x)-\mu\right)^T \Sigma^{-1}\left(f(x)-\mu\right)\right\} \prod_{j=1}^{d} |f_j'(x_j)|, \tag{28.3}$$

where the product term is a Jacobian.

Note that the density in (28.3) is not identifiable—we could scale each function by a constant, and scale the diagonal of Σ in the same way, and not change the density. To make the family identifiable we demand that f_j preserves marginal means and variances:

$$\mu_j = \mathbb{E}(Z_j) = \mathbb{E}(X_j) \;\text{ and }\; \sigma_j^2 \equiv \Sigma_{jj} = \mathrm{Var}\left(Z_j\right) = \mathrm{Var}\left(X_j\right). \tag{28.4}$$

These conditions only depend on $\mathrm{diag}(\Sigma)$ but not the full covariance matrix.

Now, let $F_j(x)$ denote the marginal distribution function of X_j. Since the component $f_j(X_j)$ is Gaussian, we have that

$$F_j(x) = \mathbb{P}\left(X_j \le x\right) = \mathbb{P}\left(Z_j \le f_j(x)\right) = \Phi\left(\frac{f_j(x) - \mu_j}{\sigma_j}\right)$$

which implies that

$$f_j(x) = \mu_j + \sigma_j \Phi^{-1}\left(F_j(x)\right). \tag{28.5}$$

The form of the density in (28.3) implies that the conditional independence graph of the nonparanormal is encoded in $\Omega = \Sigma^{-1}$, as for the parametric Normal, since the density factors with respect to the graph of Ω, and therefore obeys the global Markov property of the graph.

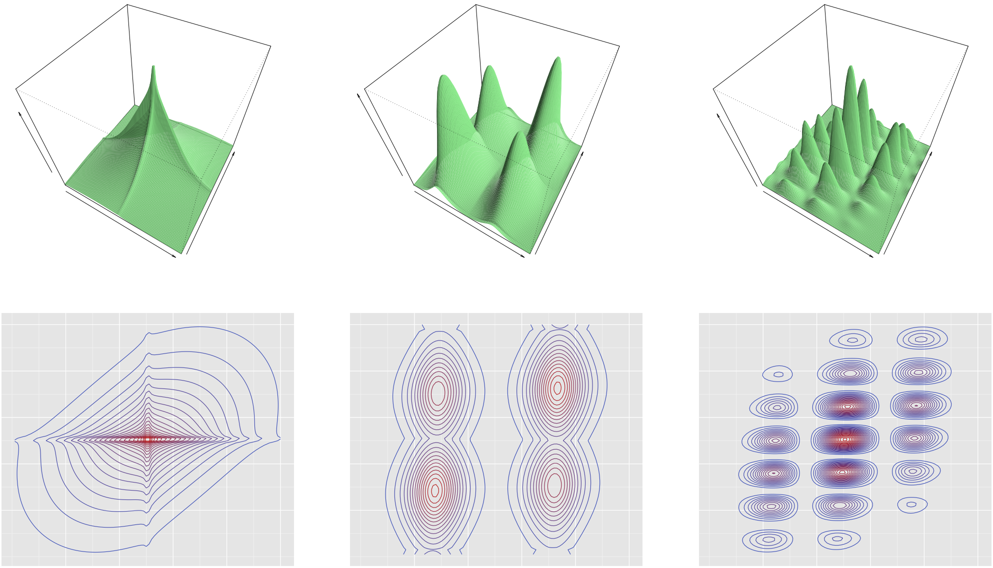


Figure 28.3. *Densities of three 2-dimensional nonparanormals. The left plots have component functions of the form $f_\alpha(x) = \text{sign}(x)|x|^\alpha$, with $\alpha_1 = 0.9$, and $\alpha_2 = 0.8$. The center plots have component functions of the form $g_\alpha(x) = \lfloor x \rfloor + 1/(1 + \exp(-\alpha(x - \lfloor x \rfloor - 1/2)))$ with $\alpha_1 = 10$ and $\alpha_2 = 5$, where $x - \lfloor x \rfloor$ is the fractional part. The right plots have component functions of the form $h_\alpha(x) = x + \sin(\alpha x)/\alpha$, with $\alpha_1 = 5$ and $\alpha_2 = 10$. In each case $\mu = (0, 0)$ and $\Sigma = \binom{1\ .5}{.5\ 1}$.*

In fact, this is true for any choice of identification restrictions; thus, it is not necessary to estimate μ or σ to estimate the graph, as the following result shows.

28.6 Lemma. *Define*

$$h_j(x) = \Phi^{-1}(F_j(x)) \tag{28.7}$$

and let Λ be the covariance matrix of $h(X)$. Then $X_j \perp\!\!\!\perp X_k \mid X_{\setminus\{j,k\}}$ if and only if $\Lambda^{-1}_{jk} =$ 0.

Proof. We can rewrite the covariance matrix as

$$\Sigma_{jk} = \text{Cov}(Z_j, Z_k) = \sigma_j \sigma_k \text{Cov}(h_j(X_j), h_k(X_k)).$$

Hence $\Sigma = D\Lambda D$ and

$$\Sigma^{-1} = D^{-1}\Lambda^{-1}D^{-1},$$

where D is the diagonal matrix with $\text{diag}(D) = \sigma$. The zero pattern of Λ^{-1} is therefore identical to the zero pattern of Σ^{-1}. □

Figure 28.3 shows three examples of 2-dimensional nonparanormal densities. The component functions are taken to be from three different families of monotonic functions—one using power transforms, one using logistic transforms, and another using sinusoids:

$$\begin{aligned} f_\alpha(x) &= \operatorname{sign}(x)|x|^\alpha \\ g_\alpha(x) &= \lfloor x \rfloor + \frac{1}{1+\exp\left\{-\alpha(x - \lfloor x \rfloor - \frac{1}{2})\right\}} \\ h_\alpha(x) &= x + \frac{\sin(\alpha x)}{\alpha}. \end{aligned}$$

The covariance in each case is $\Sigma = \binom{1\ .5}{.5\ 1}$ and the mean is $\mu = (0, 0)$. It can be seen how the concavity and number of modes of the density can change with different nonlinearities. Clearly the nonparanormal family is much richer than the Normal family.

The assumption that $f(X) = (f_1(X_1), \ldots, f_d(X_d)$ is Normal leads to a semiparametric model where only one dimensional functions need to be estimated. But the monotonicity of the functions f_j, which map onto $\mathbb{R}$, enables computational tractability of the nonparanormal. For more general functions f, the normalizing constant for the density

$$p_X(x) \propto \exp\left\{-\frac{1}{2}\left(f(x) - \mu\right)^T \Sigma^{-1} \left(f(x) - \mu\right)\right\} \tag{28.8}$$

cannot be computed in closed form.

28.3.1 Connection to Copula

If F_j is the distribution of X_j, then $U_j = F_j(X_j)$ is uniformly distributed on $(0, 1)$. Let C denote the joint distribution function of $U = (U_1, \ldots, U_d)$, and let F denote the distribution function of X. Then we have that

$$\begin{aligned} F(x_1, \ldots, x_d) &= \mathbb{P}(X_1 \le x_1, \ldots, X_d \le x_d) && (28.9) \\ &= \mathbb{P}(F_1(X_1) \le F_1(x_1), \ldots, F_d(X_d) \le F_d(x_d)) && (28.10) \\ &= \mathbb{P}(U_1 \le F_1(x_1), \ldots, U_d \le F_d(x_d)) && (28.11) \\ &= C(F_1(x_1), \ldots, F_d(x_d)). && (28.12) \end{aligned}$$

This is known as Sklar's theorem (Sklar, 1959), and C is called a *copula*. If c is the density function of C then

$$p(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{j=1}^{d} p(x_j) \tag{28.13}$$

where $p(x_j)$ is the marginal density of X_j. For the nonparanormal we have

$$F(x_1, \ldots, x_d) = \Phi_{\mu,\Sigma}(\Phi^{-1}(F_1(x_1)), \ldots, \Phi^{-1}(F_d(x_d))) \tag{28.14}$$

where $\Phi_{\mu,\Sigma}$ is the multivariate Gaussian cdf and Φ is the univariate standard Gaussian cdf.

The Gaussian copula is usually expressed in terms of the correlation matrix, which is given by $R = \operatorname{diag}(\sigma)^{-1}\Sigma \operatorname{diag}(\sigma)^{-1}$. Note that the univariate marginal density for a

Normal can be written as $p(x_j) = \frac{1}{\sigma_j}\phi(u_j)$ where $u_j = (x_j - \mu_j)/\sigma_j$. The multivariate Normal density can thus be expressed as

$$p_{\mu,\Sigma}(x_1, \ldots, x_d) = \frac{1}{(2\pi)^{d/2}|R|^{1/2}\prod_{j=1}^{d}\sigma_j}\exp\left(-\frac{1}{2}u^T R^{-1} u\right) \tag{28.15}$$

$$= \frac{1}{|R|^{1/2}}\exp\left(-\frac{1}{2}u^T(R^{-1} - I)u\right)\prod_{j=1}^{d}\frac{\phi(u_j)}{\sigma_j}. \tag{28.16}$$

Since the distribution F_j of the jth variable satisfies $F_j(x_j) = \Phi((x_j - \mu_j)/\sigma_j) = \Phi(u_j)$, we have that $(X_j - \mu_j)/\sigma_j \overset{d}{=} \Phi^{-1}(F_j(X_j))$. The Gaussian copula density is thus

$$c(F_1(x_1), \ldots, F_d(x_d)) = \frac{1}{|R|^{1/2}}\exp\Big\{-\frac{1}{2}\Phi^{-1}(F(x))^T(R^{-1} - I)\Phi^{-1}(F(x))\Big\} \tag{28.17}$$

where $\Phi^{-1}(F(x)) = (\Phi^{-1}(F_1(x_1)), \ldots, \Phi^{-1}(F_d(x_d)))$. This is seen to be equivalent to (28.3) using the chain rule and the identity

$$(\Phi^{-1})'(\eta) = \frac{1}{\phi\left(\Phi^{-1}(\eta)\right)}. \tag{28.18}$$

28.3.2 Estimation

Let $X^{(1)}, \ldots, X^{(n)}$ be a sample of size n where $X^{(i)} = (X_1^{(i)}, \ldots, X_d^{(i)})^T \in \mathbb{R}^d$. We'll design a two-step estimation procedure where first the functions f_j are estimated, and then the inverse covariance matrix Ω is estimated, after transforming to approximately Normal.

In light of (28.7) we define

$$\widehat{h}_j(x) = \Phi^{-1}(\widetilde{F}_j(x)) \tag{28.19}$$

where $\widetilde{F}_j$ is an estimator of F_j. A natural candidate for $\widetilde{F}_j$ is the marginal empirical distribution function

$$\widehat{F}_j(t) \equiv \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{\left\{X_j^{(i)} \le t\right\}}.$$

However, in this case $\widehat{h}_j(x)$ blows up at the largest and smallest values of $X_j^{(i)}$. For the high dimensional setting where n is small relative to d, an attractive alternative is to use a truncated or *Winsorized*[28] estimator:

$$\widetilde{F}_j(x) = \begin{cases} \delta_n & \text{if } \widehat{F}_j(x) < \delta_n \\ \widehat{F}_j(x) & \text{if } \delta_n \le \widehat{F}_j(x) \le 1 - \delta_n \\ (1 - \delta_n) & \text{if } \widehat{F}_j(x) > 1 - \delta_n, \end{cases} \tag{28.20}$$

[28] After Charles P. Winsor, the statistician whom John Tukey credited with his conversion from topology to statistics (Mallows, 1990).

where δ_n is a truncation parameter. There is a bias-variance tradeoff in choosing δ_n; increasing δ_n increases the bias while it decreases the variance.

Given this estimate of the distribution of variable X_j, we then estimate the transformation function f_j by

$$\widetilde{f}_j(x) \equiv \widehat{\mu}_j + \widehat{\sigma}_j \widetilde{h}_j(x) \tag{28.21}$$

where

$$\widetilde{h}_j(x) = \Phi^{-1}\left(\widetilde{F}_j(x)\right) \tag{28.22}$$

and $\widehat{\mu}_j$ and $\widehat{\sigma}_j$ are the sample mean and standard deviation:

$$\widehat{\mu}_j \equiv \frac{1}{n}\sum_{i=1}^{n} X_j^{(i)} \text{ and } \widehat{\sigma}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(X_j^{(i)} - \widehat{\mu}_j\right)^2}.$$

Now, let $S_n(\widetilde{f})$ be the sample covariance matrix of $\widetilde{f}(X^{(1)}), \ldots, \widetilde{f}(X^{(n)})$; that is,

$$\begin{aligned} S_n(\widetilde{f}) &\equiv \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{f}(X^{(i)}) - \mu_n(\widetilde{f})\right)\left(\widetilde{f}(X^{(i)}) - \mu_n(\widetilde{f})\right)^T \\ \mu_n(\widetilde{f}) &\equiv \frac{1}{n}\sum_{i=1}^{n} \widetilde{f}(X^{(i)}). \end{aligned} \tag{28.23}$$

We then estimate Ω using $S_n(\widetilde{f})$. For instance, the maximum likelihood estimator is $\widehat{\Omega}_n^{\mathrm{MLE}} = S_n(\widetilde{f})^{-1}$.

The ℓ_1-regularized estimator is

$$\widehat{\Omega}_n = \arg\min_{\Omega}\left\{\operatorname{tr}\left(\Omega S_n(\widetilde{f})\right) - \log|\Omega| + \lambda\|\Omega\|_1\right\} \tag{28.24}$$

where λ is a regularization parameter, and $\|\Omega\|_1 = \sum_{j=1}^{d}\sum_{k=1}^{d}|\Omega_{jk}|$. The estimated graph is then $\widehat{E}_n = \{(j,k) : \widehat{\Omega}_{jk} \neq 0\}$.

Thus, we use a two-step procedure to estimate the graph.

1. Replace the observations, for each variable, by their respective Normal scores, subject to a Winsorized truncation.
2. Apply the graphical lasso to the transformed data to estimate the undirected graph.

The first step is non-iterative and computationally efficient. The truncation parameter δ_n is chosen to be

$$\delta_n = \frac{1}{4n^{1/4}\sqrt{\pi\log n}} \tag{28.25}$$

and does not need to be tuned. As will be shown in Theorem 28.26, such a choice makes the nonparanormal amenable to theoretical analysis.

28.3.3 Statistical Properties of $S_n(\widetilde{f})$

The main technical result is an analysis of the covariance of the Winsorized estimator above. In particular, we show that under appropriate conditions,

$$\max_{j,k} \left| S_n(\widetilde{f})_{jk} - S_n(f)_{jk} \right| = O_P\left(\sqrt{\frac{\log d + \log^2 n}{n^{1/2}}}\right)$$

where $S_n(\widetilde{f})_{jk}$ denotes the (j,k) entry of the matrix $S_n(\widetilde{f})$. This result allows us to leverage the significant body of theory on the graphical lasso (Rothman et al., 2008; Ravikumar et al., 2009b) which we apply in step two.

28.26 Theorem. *Suppose that $d = n^{\xi}$ and let $\widetilde{f}$ be the Winsorized estimator defined in* (28.21) *with $\delta_n = \dfrac{1}{4n^{1/4}\sqrt{\pi \log n}}$. Define*

$$C(M,\xi) \equiv \frac{48}{\sqrt{\pi\xi}}\left(\sqrt{2M}-1\right)(M+2)$$

for $M, \xi > 0$. Then for any $\epsilon \geq C(M,\xi)\sqrt{\dfrac{\log d + \log^2 n}{n^{1/2}}}$ and sufficiently large n, we have

$$\mathbb{P}\left(\max_{jk}\left| S_n(\widetilde{f})_{jk} - S_n(f)_{jk} \right| > \epsilon\right) \leq \frac{c_1 d}{(n\epsilon^2)^{2\xi}} + \frac{c_2 d}{n^{M\xi - 1}} + c_3 \exp\left(-\frac{c_4 n^{1/2}\epsilon^2}{\log d + \log^2 n}\right),$$

where c_1, c_2, c_3, c_4 are positive constants.

The proof of this result involves a detailed Gaussian tail analysis, and is given in Liu et al. (2009).

Using Theorem 28.26 and the results of Rothman et al. (2008) it can then be shown that the precision matrix is estimated at the following rates in the Frobenius norm and the ℓ_2-operator norm.

$$\|\widehat{\Omega}_n - \Omega_0\|_{\mathrm{F}} = O_P\left(\sqrt{\frac{(s+d)\log d + \log^2 n}{n^{1/2}}}\right)$$

and

$$\|\widehat{\Omega}_n - \Omega_0\|_2 = O_P\left(\sqrt{\frac{s\log d + \log^2 n}{n^{1/2}}}\right),$$

where

$$s \equiv \text{Card}\left(\{(i,j) \in \{1,\ldots,d\} \times \{1,\ldots,d\} \mid \Omega_0(i,j) \neq 0,\ i \neq j\}\right)$$

is the number of nonzero off-diagonal elements of the true precision matrix.

Using the results of Ravikumar et al. (2009b), it can also be shown, under appropriate conditions, that the sparsity pattern of the precision matrix is estimated accurately with high probability. In particular, the nonparanormal estimator $\widehat{\Omega}_n$ satisfies

$$\mathbb{P}\left(\mathcal{G}\left(\widehat{\Omega}_n, \Omega_0\right)\right) \geq 1 - o(1)$$

where $\mathcal{G}(\widehat{\Omega}_n, \Omega_0)$ is the event

$$\left\{\text{sign}\left(\widehat{\Omega}_n(j,k)\right) = \text{sign}\left(\Omega_0^{-1}(j,k)\right), \quad \forall j,k \in \{1,\ldots,d\}\right\}.$$

We refer to Liu et al. (2009) for the details of the conditions and proofs.

28.4 Forest Density Estimation

We now describe a very different, but equally flexible and useful approach. Rather than assuming a transformation to normality and an arbitrary undirected graph, we restrict the graph to be a tree or forest, but allow arbitrary nonparametric distributions.

Let $p^*(x)$ be a probability density with respect to Lebesgue measure $\mu(\cdot)$ on $\mathbb{R}^d$ and let $X^{(1)},\ldots,X^{(n)}$ be n independent identically distributed $\mathbb{R}^d$-valued data vectors sampled from $p^*(x)$ where $X^{(i)} = (X_1^{(i)},\ldots,X_d^{(i)})$. Let $\mathcal{X}_j$ denote the range of $X_i^{(j)}$ and let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$.

A graph is a forest if it is acyclic. If F is a d-node undirected forest with vertex set $V_F = \{1,\ldots,d\}$ and edge set $E_F \subset \{1,\ldots,d\} \times \{1,\ldots,d\}$, the number of edges satisfies $|E_F| < d$. We say that a probability density function $p(x)$ is *supported by a forest* F if the density can be written as

$$p_F(x) = \prod_{(i,j)\in E_F} \frac{p(x_i,x_j)}{p(x_i)\,p(x_j)} \prod_{k\in V_F} p(x_k), \tag{28.27}$$

where each $p(x_i,x_j)$ is a bivariate density on $\mathcal{X}_i \times X_j$, and each $p(x_k)$ is a univariate density on $\mathcal{X}_k$.

Let $\mathcal{F}_d$ be the family of forests with d nodes, and let $\mathcal{P}_d$ be the corresponding family of densities:

$$\mathcal{P}_d = \left\{p \geq 0 : \int_{\mathcal{X}} p(x)\,d\mu(x) = 1, \text{ and } p(x) \text{ satisfies (28.27) for some } F \in \mathcal{F}_d\right\}. \tag{28.28}$$

Define the oracle forest density

$$q^* = \arg\min_{q \in \mathcal{P}_d} D(p^* \| \, q) \tag{28.29}$$

where the Kullback-Leibler divergence $D(p\| \, q)$ between two densities p and q is

$$D(p\| \, q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx, \tag{28.30}$$

under the convention that $0 \log(0/q) = 0$, and $p \log(p/0) = \infty$ for $p \neq 0$. The following is straightforward to prove.

28.31 Proposition. *Let q^* be defined as in* (28.29). *There exists a forest $F^* \in \mathcal{F}_d$, such that*

$$q^* = p^*_{F^*} = \prod_{(i,j) \in E_{F^*}} \frac{p^*(x_i, x_j)}{p^*(x_i)\, p^*(x_j)} \prod_{k \in V_{F^*}} p^*(x_k) \tag{28.32}$$

where $p^*(x_i, x_j)$ and $p^*(x_i)$ are the bivariate and univariate marginal densities of p^*.

For any density $q(x)$, the negative log-likelihood risk $R(q)$ is defined as

$$R(q) = -\mathbb{E} \log q(X) = -\int_{\mathcal{X}} p^*(x) \log q(x)\, dx. \tag{28.33}$$

It is straightforward to see that the density q^* defined in (28.29) also minimizes the negative log-likelihood loss:

$$q^* = \arg\min_{q \in \mathcal{P}_d} D(p^* \| \, q) = \arg\min_{q \in \mathcal{P}_d} R(q) \tag{28.34}$$

We thus define the oracle risk as $R^* = R(q^*)$. Using Proposition 28.31 and equation (28.27), we have

$$\begin{aligned} R^* &= R(q^*) \;=\; R(p^*_{F^*}) \\ &= -\int_{\mathcal{X}} p^*(x) \Bigg(\sum_{(i,j) \in E_{F^*}} \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} + \sum_{k \in V_{F^*}} \log\left(p^*(x_k)\right) \Bigg) dx \\ &= -\sum_{(i,j) \in E_{F^*}} I(X_i; X_j) + \sum_{k \in V_{F^*}} H(X_k), \end{aligned} \tag{28.35}$$

where

$$I(X_i; X_j) = \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i)\, p^*(x_j)}\, dx_i dx_j \tag{28.36}$$

is the mutual information between the pair of variables X_i, X_j and

$$H(X_k) = -\int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k)\, dx_k \tag{28.37}$$

is the entropy.

28.4.1 A Two-Step Procedure

If the true density $p^*(x)$ were known, by Proposition 28.31, the density estimation problem would be reduced to finding the best forest structure F_d^*, satisfying

$$F_d^* = \operatorname*{arg\,min}_{F \in \mathcal{F}_d} R(p_F^*) = \operatorname*{arg\,min}_{F \in \mathcal{F}_d} D(p^* \| \, p_F^*). \tag{28.38}$$

The optimal forest F_d^* can be found by minimizing the right hand side of (28.35). Since the entropy term $H(X) = \sum_k H(X_k)$ is constant across all forests, this can be recast as the problem of finding the maximum weight spanning forest for a weighted graph, where the weight of the edge connecting nodes i and j is $I(X_i; X_j)$. Kruskal's algorithm (Kruskal, 1956) is a greedy algorithm that is guaranteed to find a maximum weight spanning tree of a weighted graph. In the setting of density estimation, this procedure was proposed by Chow and Liu (1968) as a way of constructing a tree approximation to a distribution. At each stage the algorithm adds an edge connecting that pair of variables with maximum mutual information among all pairs not yet visited by the algorithm, if doing so does not form a cycle. When stopped early, after $k < d-1$ edges have been added, it yields the best k-edge weighted forest.

Of course, the above procedure is not practical since the true density $p^*(x)$ is unknown. We replace the population mutual information $I(X_i; X_j)$ in (28.35) by a plug-in estimate $\widehat{I}_n(X_i; X_j)$, defined as

$$\widehat{I}_n(X_i; X_j) = \int_{\mathcal{X}_i \times \mathcal{X}_j} \widehat{p}_n(x_i, x_j) \log \frac{\widehat{p}_n(x_i, x_j)}{\widehat{p}_n(x_i)\, \widehat{p}_n(x_j)}\, dx_i dx_j \tag{28.39}$$

where $\widehat{p}_n(x_i, x_j)$ and $\widehat{p}_n(x_i)$ are bivariate and univariate kernel density estimates. Given this estimated mutual information matrix $\widehat{M}_n = \left[\widehat{I}_n(X_i; X_j)\right]$, we can then apply Kruskal's algorithm (equivalently, the Chow-Liu algorithm) to find the best tree structure $\widehat{F}_n$.

Since the number of edges of $\widehat{F}_n$ controls the number of degrees of freedom in the final density estimator, an automatic data-dependent way to choose it is needed. We adopt the following two-stage procedure. First, we randomly split the data into two sets $\mathcal{D}_1$ and $\mathcal{D}_2$ of sizes n_1 and n_2; we then apply the following steps:

1. Using $\mathcal{D}_1$, construct kernel density estimates of the univariate and bivariate marginals and calculate $\widehat{I}_{n_1}(X_i; X_j)$ for $i, j \in \{1, \ldots, d\}$ with $i \neq j$. Construct a full tree $\widehat{F}_{n_1}^{(d-1)}$ with $d-1$ edges, using the Chow-Liu algorithm.

2. Using $\mathcal{D}_2$, prune the tree $\widehat{F}_{n_1}^{(d-1)}$ to find a forest $\widehat{F}_{n_1}^{(\widehat{k})}$ with $\widehat{k}$ edges, for $0 \leq \widehat{k} \leq d-1$.

Once $\widehat{F}_{n_1}^{(\widehat{k})}$ is obtained in Step 2, we can calculate $\widehat{p}_{\widehat{F}_{n_1}^{(\widehat{k})}}$ according to (28.27), using the kernel density estimates constructed in Step 1.

Step 1: Constructing a sequence of forests

Step 1 is carried out on the dataset $\mathcal{D}_1$. Let $K(\cdot)$ be a univariate kernel function. Given an evaluation point (x_i, x_j), the bivariate kernel density estimate for (X_i, X_j) based on the observations $\{X_i^{(s)}, X_j^{(s)}\}_{s\in\mathcal{D}_1}$ is defined as

$$\widehat{p}_{n_1}(x_i, x_j) = \frac{1}{n_1}\sum_{s\in\mathcal{D}_1}\frac{1}{h_2^2}K\left(\frac{X_i^{(s)}-x_i}{h_2}\right)K\left(\frac{X_j^{(s)}-x_j}{h_2}\right), \tag{28.40}$$

where we use a product kernel with $h_2 > 0$ as the bandwidth parameter. The univariate kernel density estimate $\widehat{p}_{n_1}(x_k)$ for X_k is

$$\widehat{p}_{n_1}(x_k) = \frac{1}{n_1}\sum_{s\in\mathcal{D}_1}\frac{1}{h_1}K\left(\frac{X_k^{(s)}-x_k}{h_1}\right), \tag{28.41}$$

where $h_1 > 0$ is the univariate bandwidth.

We assume that the data lie in a d-dimensional unit cube $\mathcal{X} = [0,1]^d$. To calculate the empirical mutual information $\widehat{I}_{n_1}(X_i; X_j)$, we need to numerically evaluate a two-dimensional integral. To do so, we calculate the kernel density estimates on a grid of points. We choose m evaluation points on each dimension, $x_{1i} < x_{2i} < \cdots < x_{mi}$ for the ith variable. The mutual information $\widehat{I}_{n_1}(X_i; X_j)$ is then approximated as

$$\widehat{I}_{n_1}(X_i; X_j) = \frac{1}{m^2}\sum_{k=1}^{m}\sum_{\ell=1}^{m}\widehat{p}_{n_1}(x_{ki}, x_{\ell j})\log\frac{\widehat{p}_{n_1}(x_{ki}, x_{\ell j})}{\widehat{p}_{n_1}(x_{ki})\,\widehat{p}_{n_1}(x_{\ell j})}. \tag{28.42}$$

The approximation error can be made arbitrarily small by choosing m sufficiently large. As a practical concern, care needs to be taken that the factors $\widehat{p}_{n_1}(x_{ki})$ and $\widehat{p}_{n_1}(x_{\ell j})$ in the denominator are not too small; a truncation procedure can be used to ensure this. Once the $d \times d$ mutual information matrix $\widehat{M}_{n_1} = \left[\widehat{I}_{n_1}(X_i; X_j)\right]$ is obtained, we can apply the Chow-Liu (Kruskal) algorithm to find a maximum weight spanning tree.

Tree Construction (Kruskal/Chow-Liu)

```
Input: Data set $\mathcal{D}_1$ and the bandwidths $h_1$, $h_2$.

Initialize: Calculate $\widehat{M}_{n_1}$, according to (28.40), (28.41), and (28.42).

Set $E^{(0)} = \emptyset$

For $k = 1, \dots, d-1$:

    1. Set $(i^{(k)}, j^{(k)}) \leftarrow \arg\max_{(i,j)} \widehat{M}_{n_1}(i,j)$ such that $E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$
       does not contain a cycle;
    2. $E^{(k)} \leftarrow E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$.

Output: tree $\widehat{F}_{n_1}^{(d-1)}$ with edge set $E^{(d-1)}$.
```

Step 2: Selecting a forest size

The full tree $\widehat{F}_{n_1}^{(d-1)}$ obtained in Step 1 might have high variance when the dimension d is large, leading to overfitting in the density estimate. In order to reduce the variance, we prune the tree; that is, we choose an unconnected tree with k edges. The number of edges k is a tuning parameter that induces a bias-variance tradeoff.

In order to choose k, note that in stage k of the Chow-Liu algorithm we have an edge set $E^{(k)}$ (in the notation of the Algorithm 28.4.1) which corresponds to a forest $\widehat{F}_{n_1}^{(k)}$ with k edges, where $F_{n_1}^{(0)}$ is the union of d disconnected nodes. To select k, we cross-validate over the d forests $\widehat{F}_{n_1}^{(0)}$, $\widehat{F}_{n_1}^{(1)}$, ... , $\widehat{F}_{n_1}^{(d-1)}$.

Let $\widehat{p}_{n_2}(x_i, x_j)$ and $\widehat{p}_{n_2}(x_k)$ be defined as in (28.40) and (28.41), but now evaluated solely based on the held-out data in $\mathcal{D}_2$. For a density p_F that is supported by a forest F, we define the held-out negative log-likelihood risk as

$$\begin{aligned}&\widehat{R}_{n_2}(p_F) \qquad (28.43)\\ &\quad = -\sum_{(i,j)\in E_F} \int_{\mathcal{X}_i \times \mathcal{X}_j} \widehat{p}_{n_2}(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)\, p(x_j)}\, dx_i dx_j - \sum_{k \in V_F} \int_{\mathcal{X}_k} \widehat{p}_{n_2}(x_k) \log p(x_k)\, dx_k.\end{aligned}$$

The selected forest is then $\widehat{F}_{n_1}^{(\widehat{k})}$ where

$$\widehat{k} = \mathop{\arg\min}_{k \in \{0, \dots, d-1\}} \widehat{R}_{n_2}\left(\widehat{p}_{F_{n_1}^{(k)}}\right) \qquad (28.44)$$

and where $\widehat{p}_{F_{n_1}^{(k)}}$ is computed using the density estimate $\widehat{p}_{n_1}$ constructed on $\mathcal{D}_1$.

We can also estimate $\widehat{k}$ as

$$\widehat{k} = \underset{k\in\{0,\ldots,d-1\}}{\arg\max} \; \frac{1}{n_2}\sum_{s\in\mathcal{D}_2} \log\left(\prod_{(i,j)\in E_{F^{(k)}}} \frac{\widehat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\widehat{p}_{n_1}(X_i^{(s)})\,\widehat{p}_{n_1}(X_j^{(s)})} \prod_{\ell\in V_{F^{(k)}}} \widehat{p}_{n_1}(X_\ell^{(s)})\right) \tag{28.45}$$

$$= \underset{k\in\{0,\ldots,d-1\}}{\arg\max} \; \frac{1}{n_2}\sum_{s\in\mathcal{D}_2} \log\left(\prod_{(i,j)\in E_{F^{(k)}}} \frac{\widehat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\widehat{p}_{n_1}(X_i^{(s)})\,\widehat{p}_{n_1}(X_j^{(s)})}\right). \tag{28.46}$$

This minimization can be efficiently carried out by iterating over the $d-1$ edges in $\widehat{F}_{n_1}^{(d-1)}$.

Once $\widehat{k}$ is obtained, the final forest-based kernel density estimate is given by

$$\widehat{p}_n(x) = \prod_{(i,j)\in E^{(\widehat{k})}} \frac{\widehat{p}_{n_1}(x_i, x_j)}{\widehat{p}_{n_1}(x_i)\,\widehat{p}_{n_1}(x_j)} \prod_k \widehat{p}_{n_1}(x_k). \tag{28.47}$$

Another alternative is to compute a maximum weight spanning forest, using Kruskal's algorithm, but with heldout edge weights

$$\widehat{w}_{n_2}(i,j) = \frac{1}{n_2}\sum_{s\in\mathcal{D}_2} \log \frac{\widehat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\widehat{p}_{n_1}(X_i^{(s)})\,\widehat{p}_{n_1}(X_j^{(s)})}. \tag{28.48}$$

In fact, asymptotically (as $n_2 \to \infty$) this gives optimal tree-based estimator constructed in terms of the kernel density estimates $\widehat{p}_{n_1}$.

28.4.2 Statistical Properties

The statistical properties of forest density estimator can be analyzed under the same type of assumptions that are made for classical kernel density estimation. In particular, assume that the univariate and bivariate densities lie in a Hölder class with exponent β. Under this assumption the minimax rate of convergence in the squared error loss is $O(n^{\beta/(\beta+1)})$ for bivariate densities and $O(n^{2\beta/(2\beta+1)})$ for univariate densities. Technical assumptions on the kernel yield L_∞ concentration results on kernel density estimation (Giné and Guillou, 2002).

Choose the bandwidths h_1 and h_2 to be used in the one-dimensional and two-dimensional kernel density estimates according to

$$h_1 \asymp \left(\frac{\log n}{n}\right)^{\frac{1}{1+2\beta}} \tag{28.49}$$

$$h_2 \asymp \left(\frac{\log n}{n}\right)^{\frac{1}{2+2\beta}}. \tag{28.50}$$

This choice of bandwidths ensures the optimal rate of convergence. Let $\mathcal{P}_d^{(k)}$ be the family of d-dimensional densities that are supported by forests with at most k edges. Then

$$\mathcal{P}_d^{(0)} \subset \mathcal{P}_d^{(1)} \subset \cdots \subset \mathcal{P}_d^{(d-1)}. \tag{28.51}$$

Due to this nesting property,

$$\inf_{q_F \in \mathcal{P}_d^{(0)}} R(q_F) \geq \inf_{q_F \in \mathcal{P}_d^{(1)}} R(q_F) \geq \cdots \geq \inf_{q_F \in \mathcal{P}_d^{(d-1)}} R(q_F). \tag{28.52}$$

This means that a full spanning tree would generally be selected if we had access to the true distribution. However, with access to finite data to estimate the densities ($\widehat{p}_{n_1}$) the optimal procedure is to use fewer than $d-1$ edges. The following result analyzes the excess risk resulting from selecting the forest based on the heldout risk $\widehat{R}_{n_2}$.

28.53 Theorem. *Let $\widehat{p}_{\widehat{F}_d^{(k)}}$ be the estimate with $|E_{\widehat{F}_d^{(k)}}| = k$ obtained after the first k iterations of the Chow-Liu algorithm. Then under (omitted) technical assumptions on the densities and kernel, for any $1 \leq k \leq d-1$,*

$$R(\widehat{p}_{\widehat{F}_d^{(k)}}) - \inf_{q_F \in \mathcal{P}_d^{(k)}} R(q_F) = O_P\left(k\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}\right) \tag{28.54}$$

and

$$R(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - \min_{0 \leq k \leq d-1} R(\widehat{p}_{\widehat{F}_d^{(k)}}) = O_P\left((k^* + \widehat{k})\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}\right) \tag{28.55}$$

where $\widehat{k} = \arg\min_{0 \leq k \leq d-1} \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_d^{(k)}})$ and $k^* = \arg\min_{0 \leq k \leq d-1} R(\widehat{p}_{\widehat{F}_d^{(k)}})$.

The main work in proving this result lies in establishing bounds such as

$$\sup_{F \in \mathcal{F}_d^{(k)}} |R(\widehat{p}_F) - \widehat{R}_{n_2}(\widehat{p}_F)| = O_P\Big(\phi_n(k) + \psi_n(d)\Big) \tag{28.56}$$

where $\widehat{R}_{n_2}$ is the heldout risk, under the notation

$$\phi_n(k) = k\sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \tag{28.57}$$

$$\psi_n(d) = d\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}. \tag{28.58}$$

For the proof of this and related results, see Liu et al. (2011). Using this, one easily obtains

$$\begin{aligned} R(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - R(\widehat{p}_{\widehat{F}_d^{(k^*)}}) &= R(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) + \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - R(\widehat{p}_{\widehat{F}_d^{(k^*)}}) && (28.59) \\ &= O_P(\phi_n(\widehat{k}) + \psi_n(d)) + \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - R(\widehat{p}_{\widehat{F}_d^{(k^*)}}) && (28.60) \\ &\leq O_P(\phi_n(\widehat{k}) + \psi_n(d)) + \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_d^{(k^*)}}) - R(\widehat{p}_{\widehat{F}_d^{(k^*)}}) && (28.61) \\ &= O_P\left(\phi_n(\widehat{k}) + \phi_n(k^*) + \psi_n(d)\right). && (28.62) \end{aligned}$$

source: wikipedia.org

Arabidopsis thaliana is a small flowering plant; it was the first plant genome to be sequenced, and its roughly 27,000 genes and 35,000 proteins have been actively studied. Here we consider a data set based on Affymetrix GeneChip microarrays with sample size $n = 118$, for which $p = 40$ genes have been selected for analysis.

where (28.61) follows from the fact that $\widehat{k}$ is the minimizer of $\widehat{R}_{n_2}(\cdot)$.

Note that this result allows the dimension d to increase at a rate $o\left(\sqrt{n^{2\beta/(1+2\beta)}/\log n}\right)$ and the number of edges k to increase at a rate $o\left(\sqrt{n^{\beta/(1+\beta)}/\log n}\right)$, with the excess risk still decreasing to zero asymptotically.

28.5 Examples

28.5.1 Gene-Gene Interaction Graphs

The nonparanormal and Gaussian graphical model can construct very different graphs. Here we consider a data set based on Affymetrix GeneChip microarrays for the plant *Arabidopsis thaliana*, (Wille et al., 2004). The sample size is $n = 118$. The expression levels for each chip are pre-processed by log-transformation and standardization. A subset of 40 genes from the isoprenoid pathway is chosen for analysis.

While these data are often treated as multivariate Gaussian, the nonparanormal and the glasso give very different graphs over a wide range of regularization parameters, suggesting that the nonparametric method could fffereent biological conclusions.

The regularization paths of the two methods are compared in Figure 28.4. To generate the paths, we select 50 regularization parameters on an evenly spaced grid in the interval $[0.16, 1.2]$. Although the paths for the two methods look similar, there are some subtle differences. In particular, variables become nonzero in a different order.

Figure 28.5 compares the estimated graphs for the two methods at several values of the regularization parameter λ in the range $[0.16, 0.37]$. For each λ, we show the estimated graph from the nonparanormal in the first column. In the second column we show the graph obtained by scanning the full regularization path of the glasso fit and finding the graph having the smallest symmetric difference with the nonparanormal graph. The sym-

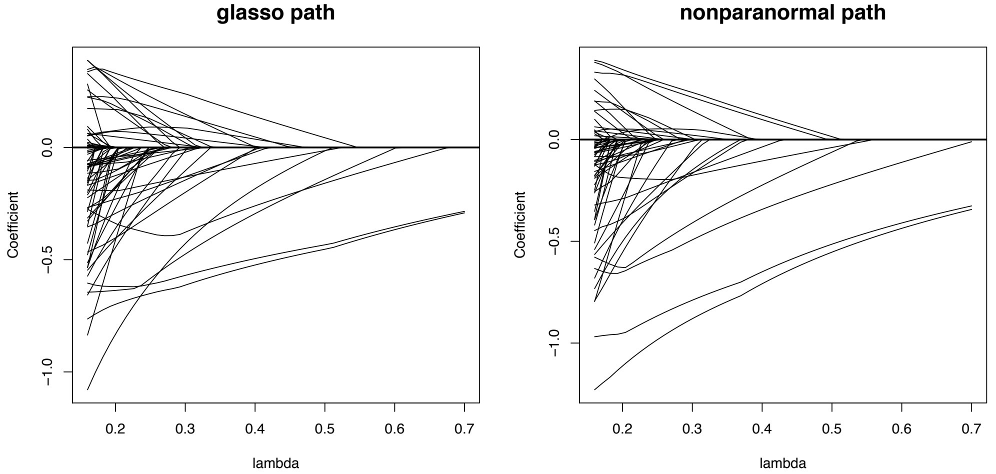


Figure 28.4. *Regularization paths of both methods on the microarray data set. Although the paths for the two methods look similar, there are some subtle differences.*

metric difference graph is shown in the third column. The closest glasso fit is different, with edges selected by the glasso not selected by the nonparanormal, and vice-versa. The estimated transformation functions for several genes are shown Figure 28.6, which show non-Gaussian behavior.

Since the graphical lasso typically results in a large parameter bias as a consequence of the ℓ_1 regularization, it sometimes make sense to use the *refit glasso*, which is a two-step procedure—in the first step, a sparse inverse covariance matrix is obtained by the graphical lasso; in the second step, a Gaussian model is refit without ℓ_1 regularization, but enforcing the sparsity pattern obtained in the first step.

Figure 28.7 compares forest density estimation to the graphical lasso and refit glasso. It can be seen that the forest-based kernel density estimator has better generalization performance. This is not surprising, given that the true distribution of the data is not Gaussian. (Note that since we do not directly compute the marginal univariate densities in the nonparanormal, we are unable to compute likelihoods under this model.) The held-out log-likelihood curve for forest density estimation achieves a maximum when there are only 35 edges in the model. In contrast, the held-out log-likelihood curves of the glasso and refit glasso achieve maxima when there are around 280 edges and 100 edges respectively, while their predictive estimates are still inferior to those of the forest-based kernel density estimator. Figure 28.7 also shows the estimated graphs for the forest-based kernel density estimator and the graphical lasso. The graphs are automatically selected based on held-out log-likelihood, and are clearly different.

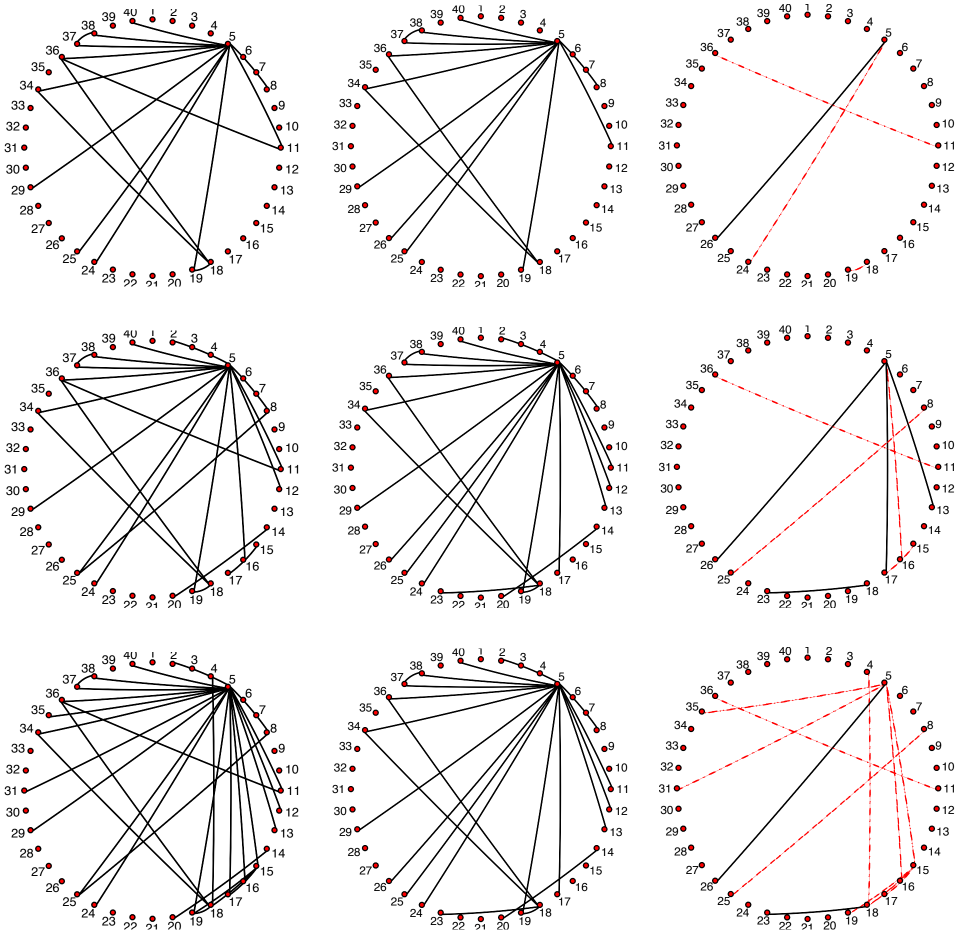


Figure 28.5. *The nonparanormal estimated graph for three values of* $\lambda = 0.2448, 0.2661, 0.30857$ *(left column), the closest glasso estimated graph from the full path (middle) and the symmetric difference graph (right).*

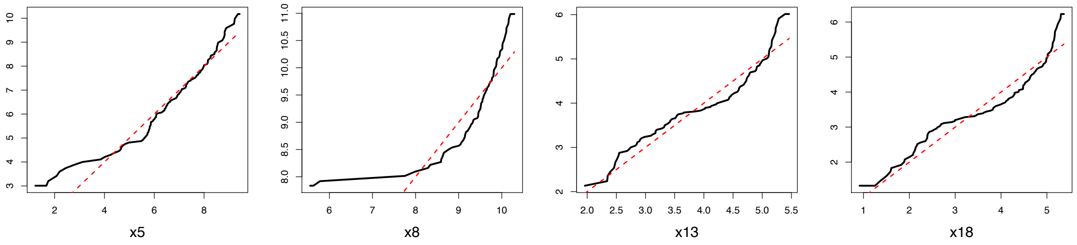


Figure 28.6. *Estimated transformation functions for four genes in the microarray data set, indicating non-Gaussian marginals. The corresponding genes are among the nodes appearing in the symmetric difference graphs above.*

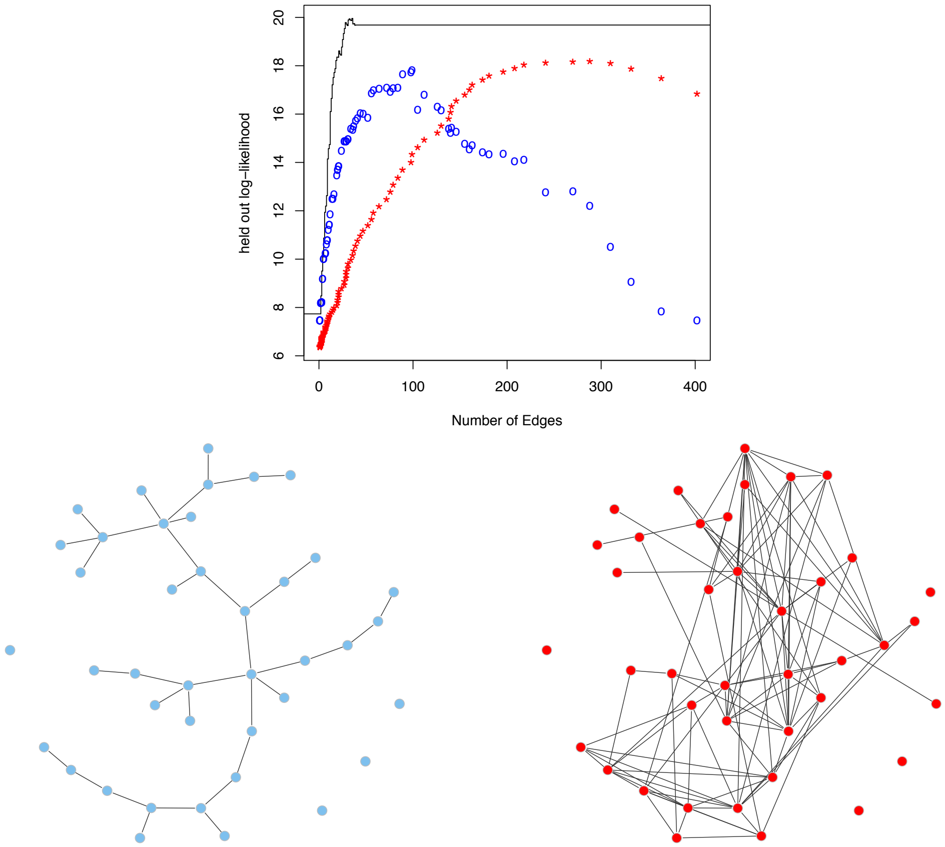
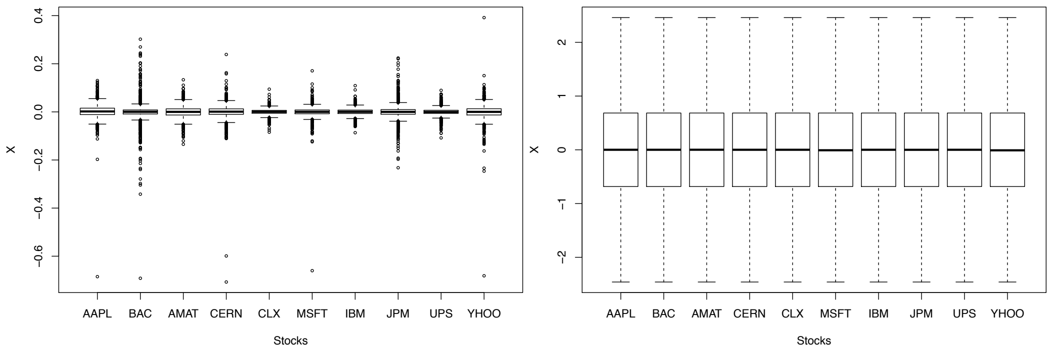


Figure 28.7. *Results on microarray data. Top: held-out log-likelihood of the forest density estimator (black step function), glasso (red stars), and refit glasso (blue circles). Bottom: estimated graphs using the forest-based estimator (left) and the glasso (right), using the same node layout.*

28.5.2 Graphs for Equities Data

For the examples in this section we collected stock price data from Yahoo! Finance (`finance.yahoo.com`). The daily closing prices were obtained for 452 stocks that consistently were in the S&P 500 index between January 1, 2003 through January 1, 2011. This gave us altogether 2,015 data points, each data point corresponds to the vector of closing prices on a trading day. With $S_{t,j}$ denoting the closing price of stock j on day t, we consider the variables $X_{tj} = \log\left(S_{t,j}/S_{t-1,j}\right)$ and build graphs over the indices j. We simply treat the instances X_t as independent replicates, even though they form a time series. We Winsorize (or truncate) every stock so that its data points are within six times the mean absolute deviation from the sample average. In Figure 28.8(a) we show boxplots for 10 randomly chosen stocks. It can be seen that the data contains outliers even after Winsorization; the reasons for these outliers includes splits in a stock, which increases the number of shares. In Figure



(a) original data

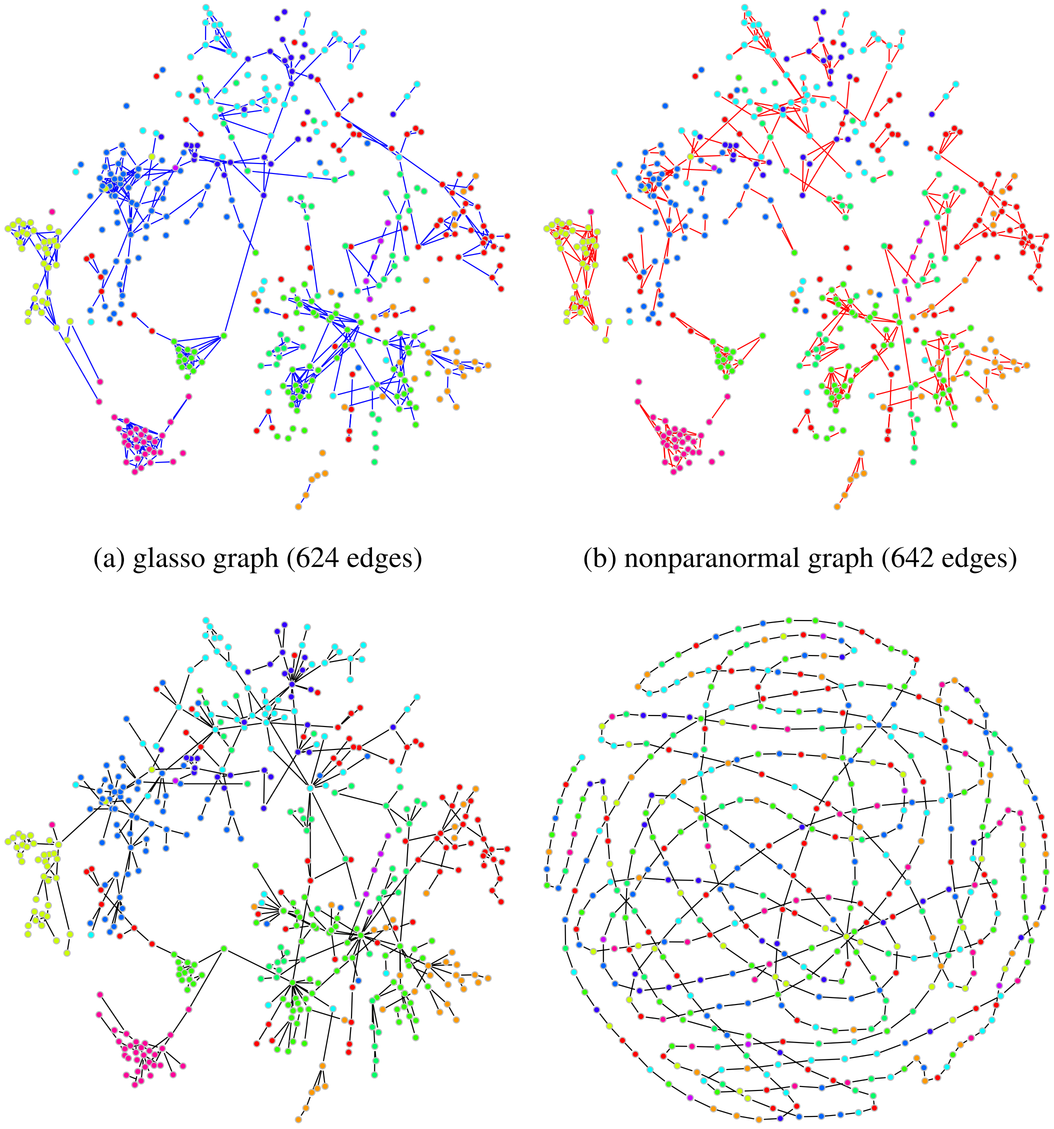
(b) after nonparanormal transformation

Figure 28.8. *Boxplots of $X_t = \log(S_t/S_{t-1})$ for 10 stocks. As can be seen, the original data has many outliers, which is addressed by the nonparanormal transformation on the re-scaled data (right).*

28.8(b) we show the boxplots of the data after the nonparanormal transformation. We show below how removing outliers is important for forest density estimation. In the results show below, we use the subset of the data between January 1, 2003 to January 1, 2008, before the onset of the "financial crisis." It is interesting to compare to results that include data after 2008, but we omit these for brevity.

The 452 stocks are categorized into 10 Global Industry Classification Standard (GICS) sectors, including `Consumer Discretionary` (70 stocks), `Consumer Staples` (35 stocks), `Energy` (37 stocks), `Financials` (74 stocks), `Health Care` (46 stocks), `Industrials` (59 stocks), `Information Technology` (64 stocks), `Materials` (29 stocks), `Telecommunications Services` (6 stocks), and `Utilities` (32 stocks). It is expected that stocks from the same GICS sectors should tend to be clustered together, since stocks from the same GICS sector tend to interact more with each other. In the graphs shown below, the nodes are colored according to the GICS sector of the corresponding stock.

In Figures 28.9(a)-(c) we show graphs estimated using the glasso, nonparanormal, and forest density estimator on the data from January 1, 2003 to January 1, 2008. There are altogether $n = 1,257$ data points and $d = 452$ dimensions. To estimate the nonparanormal graph, we adopt a variant of the stability selection method proposed by Meinshausen and Bühlmann (2010). More specifically, let $\lambda_{\max}$ be the smallest tuning parameter λ such that the estimated nonparanormal graph using Equation (28.24) is empty, and let $\widetilde{\lambda} = 0.1\lambda_{\max}$. We randomly sample 50 sub-datasets, each containing $B = \lfloor 10\sqrt{n} \rfloor = 320$ data points. On each of these 50 subsampled dataset, we estimate a nonparanormal graph using (28.24) with $\lambda = \widetilde{\lambda}$. In the final nonparanormal graph shown in Figure 28.9(b), an edge is present only if it appears more than 95 percent of the time among the 50 subsampled datasets. Therefore the nonparanormal graph is in fact a stability graph; the graph has 642 edges. To estimate the glasso graph, we again take $\lambda'_{\max}$ to be the smallest tuning parameter such that the



(a) glasso graph (624 edges)

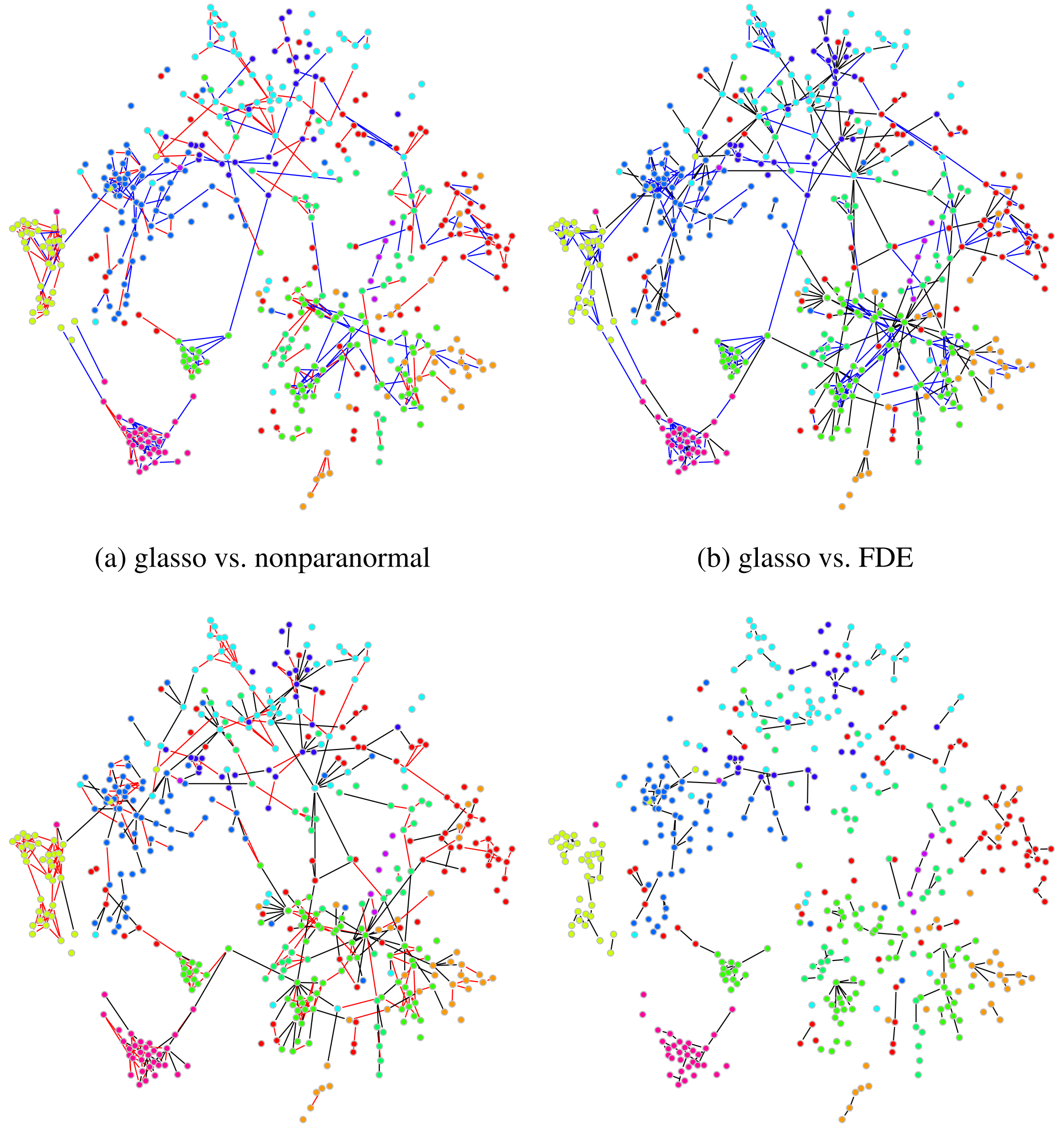
(b) nonparanormal graph (642 edges)

(c) FDE graph (451 edges)

(d) FDE graph with outliers (451 edges)

Figure 28.9. *Graphs build on S&P 500 stock data from Jan. 1, 2003 to Jan. 1, 2008. The graphs are estimated using (a) the glasso, (b) the nonparanormal, and (c) forest density estimation. The nodes are colored according to their GICS sector categories. Figure (d), shows the forest graph obtained without transforming the original data to remove outliers.*

estimated glasso graph is empty, randomly subsample 50 datasets with block size $B = 320$, and fit a glasso graph using the tuning parameter $\widetilde{\lambda}' = 0.1\lambda'_{\max}$. We then plot all the edges whose frequency of occurrence is no smaller than a threshold $\rho \in [0, 1]$, where ρ is chosen



(a) glasso vs. nonparanormal

(b) glasso vs. FDE

(c) nonparanormal vs. FDE

(d) edges common to glasso and npn

Figure 28.10. *Visualizations of the differences between the estimated graphs. The symmetric difference between the glasso and nonparanormal graphs are shown in (a) (a blue edge is unique to the glasso graph while a red edge is unique to the nonparanormal graph). The graphs in (b) and (c) similarly illustrate the symmetric difference of the glasso and FDE graphs, and of the nonparanormal and FDE graphs. Blue edges are unique to the glasso graph, red edges are unique to the nonparanormal graph, and black edges are unique to the FDE graph. The shared edges of the glasso and nonparanormal graphs are shown in (d).*

such that the total number of edges in the glasso graph is closest to the nonparanormal graph. The final estimated glasso graph has 624 edges and is shown in Figure 28.9(a).

Since the dataset contains $n = 1,257$ data points, we directly apply the forest density estimator on the whole dataset to obtain a full spanning tree of $d - 1 = 451$ edges. This estimator turns out to be very sensitive to outliers, since it exploits kernel density estimates as building blocks. In Figure 28.9(d) we show the estimated forest density graph on the stock data when outliers are *not* removed. In this case the graph is anomolous, with a snake-like character that weaves in and out of the 10 GICS industries. Intuitively, the outliers make the two-dimensional densities appear like thin “pancakes,” and densities with similar orientations are clustered together. To address this, we transform by the nonparanormal transformation, and then run forest density estimation. Figure 28.9(c) shows the estimated forest graph after outliers are removed in this way. The resulting graph has good clustering with respect to the GICS sectors.

Figures 28.10(a)-(c) display the differences between the glasso, nonparanormal, and forest density estimation graphs. Figure 28.10(d) shows the shared edges between the estimated glasso and nonparanormal graphs. Although the nonparanormal and glasso graph topologies appear similar as shown, with respect to the clustering behavior in the GICS classes, they have many different edges. In fact, the nonparanormal and glasso graphs share only about 63% of the same edges. In comparing the nonparanormal and glasso graphs with the forest density estimation graphs, we find 58.5% edges in the forest density estimation graph are also contained in the nonparanormal graph. In contrast, only 43% of edges in the forest density estimation graph are contained in the glasso graph.

We refrain from drawing any hard conclusions about the effectiveness of the different methods based on these plots—how these graphs are used will depend on the application. These results serve mainly to highlight how very different inferences about the independence relations can arise from moving from a Gaussian model to a semiparametric model to a fully nonparametric model with restricted graphs.

28.6 Discussion

This paper has considered undirected graphical models for continuous data, where the general densities take the form

$$p(x) \propto \exp\left(\sum_{C \in \text{Cliques}(G)} f_C(x_C)\right). \tag{28.63}$$

Such a general family is at least as difficult as the general high-dimensional nonparametric regression model. But, as for regression, simplifying assumptions can lead to tractable and useful models. We have considered two approaches that make very different tradeoffs between statistical generality and computational efficiency. The nonparanormal relies on estimating one-dimensional functions, in a manner that is similar to the way additive models estimate one-dimensional regression functions. This allows arbitrary graphs, but the distribution is semiparametric, via the Gaussian copula. At the other extreme, when we restrict to acyclic graphs we can have fully nonparametric bivariate and univariate marginals. This leverages classical techniques for low-dimensional density estimation, together with approximation algorithms for constructing the graph. Clearly these are just two among many possibilities for nonparametric graphical modeling. We conclude, then, with a brief description of a few potential directions for future work.

As we saw with the nonparanormal, if only the graph is of interest, it may not be important to estimate the functions accurately. More generally, to estimate the graph it is not necessary to estimate the density. One of the most effective and theoretically well-supported methods for estimating Gaussian graphs is due to Meinshausen and Bühlmann (2006). In this approach, we regress each variable X_j onto all other variables $(X_k)_{k \neq j}$ using the lasso. This directly estimates the set of neighbors $\mathcal{N}(j) = \{k \mid (j,k) \in E\}$ for each node j in the graph, but the covariance matrix is *not* directly estimated. Lasso theory gives conditions and guarantees on these variable selection problems. This approach was adapted to the discrete case by Ravikumar et al. (2010a), where the normalizing constant and thus the density can't be efficiently computed. This general strategy may be attractive for graph selection in nonparametric graphical models. In particular, each variable could be regressed on the others using a nonparametric regression method that performs variable selection; one such method with theoretical guarantees is due to Lafferty and Wasserman (2008).

No matter how the methodology develops, nonparametric graphical models will at best be approximations to the true distribution in many applications. Yet, there is plenty of experience to show how incorrect models can be useful. An ongoing challenge in nonparametric graphical modeling will be to better understand how the structure can be accurately estimated even when the model is wrong.

28.7 Nonparametric Belief Propagation

The nonparanormal and forest densities are special classes of tractable nonparametric graphical models. More generally, we would like to be able to work with models of the form

$$p(x) = \frac{1}{Z(f)} \exp\left(\sum_C f_C(x_C)\right) \tag{28.64}$$

as already discussed above. Doing so requires approximate inference, for example stochastic simulation or variational methods. *Nonparametric belief propagation* is a hybrid of variational approximation and simulation that has been proposed for working with general nonparametric graphical models with continuous variables.

Suppose we are given the functions f_C. How do we carry out inference for this model? In machine learning parlance, this means, for example, computing or approximating the marginal density $p(x_i)$ for one of the component variables X_i. We will consider this question in the setting where a conditional density is specified in the form

$$p(x \mid y) \propto \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j; y) \prod_{i \in V} \psi_i(x_i; y) \tag{28.65}$$

where we condition on *evidence* y and specify the model in terms of a set of *edge potentials* $\psi_{ij}(x_i, x_j; y) > 0$ and *vertex potentials* $\psi_i(x_i; y) > 0$. This model can be thought of as a form of conditional random field. Our objective is to approximate the conditional density of a single node $p(x_i \,|\, y)$.

As indicated in Chapter 20, belief propagation is a message passing algorithm that can be used for this purpose. Although we presented it for discrete distributions, it in principle extends to continuous distributions. The messages are given by densities

$$m_{ji}(x_i) \propto \int_{\mathcal{X}_j} \psi_{ij}(x_i, x_j; y)\, \psi_j(x_j; y) \prod_{i \in \mathcal{N}(j) \backslash i} m_{kj}(x_j)\, d\mu(x_j) \tag{28.66}$$

where $\mathcal{X}_j$ is the domain of variable X_j; this is the outgoing message sent from node j to node i, given the incoming messages received by j from its other neighbors, $\mathcal{N}(j) \backslash i$. With all of the incoming messages defined, the approximation to $p(x_i \,|\, y)$ is then given by

$$q(x_i \,|\, y) \propto \psi_i(x_i; y) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i). \tag{28.67}$$

The difficulty is that the integrals required to compute the messages in (28.66) may be difficult to evaluate numerically. Nonparametric belief propagation uses particle filtering methods to approximate these integrals.

To explain the algorithm in its simplest form, suppose that the incoming messages $m_{kj}(x_j)$ have been determined, and we wish to compute the outgoing message $m_{ji}(x_i)$; this is done using the following three-step procedure.

Nonparametric Belief Propagation

1. Draw a sample of L "auxiliary particles" $\widetilde{x}_{ji}^{(\ell)}$, by stochastic sampling according to
$$\widetilde{X}_{ji}^{(\ell)} \sim b_{ji}(X_j)\, \psi_j(X_j; y) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(X_j) \tag{28.68}$$
where the bias term b_{ji} is computed as
$$b_{ji}(x_j) = \int_{\mathcal{X}_i} \psi_{ij}(x_i, x_j; y)\, d\mu(x_i). \tag{28.69}$$
The sample is formed using importance sampling or another standard MCMC procedure.

2. Given the auxiliary particles, draw a sample of L particles $x_{ji}^{(\ell)}$ by sampling according to
$$X_{ji}^{(\ell)} \sim \psi_{ij}(X_i, \widetilde{x}_{ji}^{(\ell)}; y). \tag{28.70}$$

3. Given the particles $x_{ji}^{(\ell)}$, the message $m_{ji}(x_i)$ is obtained as the kernel density estimate
$$m_{ji}(x_i) = \frac{1}{L} \sum_{\ell=1}^{L} K_h(x_i, x_{ji}^{(\ell)}) \tag{28.71}$$
for an appropriately chosen bandwidth h.

Using the Gaussian kernel, this algorithm represents each message $m_{jk}(x_k)$ as a mixture of L Gaussians. Note then that if node j has d neighbors, the term $\prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(x_j)$ in the first step can be expressed as a mixture of L^{d-1} Gaussians. If L and d are large, sampling from this explicitly may be computationally prohibitive; a stochastic simulation algorithm will generally require $O(dL)$ cost.

The marginal densities $p(x_i \,|\, y)$ are approximated by simulation by sampling according to
$$X_i^{(\ell)} \sim \psi_i(X_i; y) \prod_{j \in \mathcal{N}(i)} m_{ji}(X_i). \tag{28.72}$$
A kernel density estimate of the marginal is then formed from the resulting particles as
$$q(x_i \,|\, y) = \frac{1}{L} \sum_{\ell=1}^{L} K_h\big(x_i, x_i^{(\ell)}\big). \tag{28.73}$$

The algorithm iterates until the messages converge. This nonparametric belief propagation procedure can thus be seen as a hybrid of variational methods and simulation. Details and variants of this procedure, together with interesting applications to visual tracking and sensor localization, are discussed by Sudderth et al. (2010).

28.8 Bibliographic Remarks

There is surprisingly little work on structure learning of nonparametric graphical models in high dimensions. One piece of related work is sparse log-density smoothing spline ANOVA models, introduced by Jeon and Lin (2006). In such a model the log-density function is decomposed as the sum of a constant term, one dimensional functions (main effects), two-dimensional functions (two-way interactions), and so on:

$$\log p(x) = f(x) \equiv c + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k) + \cdots . \tag{28.74}$$

The component functions satisfy certain constraints so that the model is identifiable. In high dimensions, the model is truncated up to second order interactions so that the computation is still tractable. There is a close connection between the log-density ANOVA model and undirected graphical models. For a model with only main effects and two-way interactions, we define a graph $G = (V, E)$ such that $(i, j) \in E$ if and only if $f_{ij} \neq 0$. It can be seen that $p(x)$ is Markov to G. Jeon and Lin (2006) assume that these component functions belong to certain reproducing kernel Hilbert spaces (RKHSs) equipped with a RKHS norm $\|\cdot\|_K$. To obtain a sparse estimation of the component functions $f(x)$, they propose a penalized M-estimator

$$\widehat{f} = \arg\max_{f} \Big\{ \frac{1}{n} \sum_{i=1}^{n} \exp\Big(f(X^{(i)}) \Big) + \int f(x)\rho(x)dx + \lambda J(f) \Big\}, \tag{28.75}$$

where $\rho(x)$ is some pre-defined positive density and $J(f)$ is a sparsity-inducing penalty that takes the form

$$J(f) = \sum_{j=1}^{d} \|f_j\|_K + \sum_{j<k} \|f_{jk}\|_K. \tag{28.76}$$

Solving (28.75) only requires one-dimensional integrals which can be efficiently computed. However, the optimization in (28.75) exploits a surrogate loss instead of the log-likelihood loss, and is more difficult to analyze theoretically.

Another related idea is to conduct structure learning using nonparametric decomposable graphical models (Schwaighofer et al., 2007). A distribution is a decomposable graphical model if it is Markov to a graph $G = (V, E)$ which has a junction tree representation, which can be viewed as an extension of tree-based graphical models. A junction tree yields a factorized form

$$p(x) = \frac{\prod_{C \in V_T} p(x_C)}{\prod_{S \in E_T} p(x_S)} \tag{28.77}$$

where V_T denotes the set of cliques in V and E_T is the set of separators, i.e., the intersection of two neighboring cliques in the junction tree. Exact search for the junction tree structure that maximizes the likelihood is usually computationally expensive. Schwaighofer et al. (2007) propose a forward-backward strategy for nonparametric structure learning. However, such a greedy procedure does not guarantee that the global optimal solution is found, and makes theoretical analysis challenging.

A different framework for nonparametricity involves conditioning on a collection of observed explanatory variables Z. Liu et al. (2010) develop a nonparametric procedure called *Graph-optimized CART*, or *Go-CART*, to estimate the graph conditionally under a Gaussian model. The main idea is to build a tree partition on the Z space just as in CART (classification and regression trees), but to estimate a graph at each leaf using the glasso. Oracle inequalities on risk minimization and model selection consistency were established for Go-CART by Liu et al. (2010). When Z is time, graph-valued regression reduces to the time-varying graph estimation problem (Zhou et al., 2010; Chen et al., 2010; Kolar et al., 2009).

In parametric settings, Chandrasekaran et al. (2010) and Choi et al. (2010) develop algorithms and theory for learning graphical models with latent variables. The first paper assumes the joint distribution of the observed and latent variables is a Gaussian graphical model, and the second paper assumes the joint distribution is discrete and factors according to a forest. Since the nonparanormal and forest density estimator are nonparametric versions of the Gaussian and forest graphical models for discrete data, we expect similar techniques to those of Chandrasekaran et al. (2010); Choi et al. (2010) can be used to extend the methods of this chapter to handle latent variables.

Exercises

28.1 Let X be a random variable with mean zero, unit variance, distribution F, and density $p(x)$ (with respect to Lebesgue measure). Show that

$$p(x) = f'(x)\,\phi(f(x)) \tag{28.78}$$

where $f(x) = \Phi^{-1}(F(x))$. Thus, any one dimensional distribution can be transformed to normal by a monotonic transformation.

28.2 Suppose that P is a distribution with density p, and that the undirected graph of P is a forest with edge set E_F and vertex set V_F. Show that

$$p_F(x) = \prod_{(i,j)\in E_F} \frac{p(x_i, x_j)}{p(x_i)\,p(x_j)} \prod_{k\in V_F} p(x_k). \tag{28.79}$$

28.3 Prove Proposition 28.31