6-2010

Forest Density Estimation

Anupam Gupta
Carnegie Mellon University, anupamg@cs.cmu.edu

John Lafferty
Carnegie Mellon University, lafferty@cs.cmu.edu

Han Liu
Carnegie Mellon University

Larry Wasserman
Carnegie Mellon University, larry@stat.cmu.edu

Min Xiu
Carnegie Mellon University

Published In
.

Forest Density Estimation

Anupam Gupta[†], **John Lafferty**[†*], **Han Liu**[†*], **Larry Wasserman**[*†], **Min Xu**[†]

[†]School of Computer Science
[*]Department of Statistics
Carnegie Mellon University

Abstract

We study graph estimation and density estimation in high dimensions, using a family of density estimators based on forest structured undirected graphical models. For density estimation, we do not assume the true distribution corresponds to a forest; rather, we form kernel density estimates of the bivariate and univariate marginals, and apply Kruskal's algorithm to estimate the optimal forest on held out data. We prove an oracle inequality on the excess risk of the resulting estimator relative to the risk of the best forest. For graph estimation, we consider the problem of estimating forests with restricted tree sizes. We prove that finding a maximum weight spanning forest with restricted tree size is NP-hard, and develop an approximation algorithm for this problem. Viewing the tree size as a complexity parameter, we then select a forest using data splitting, and prove bounds on excess risk and structure selection consistency of the procedure. Experiments with simulated data and microarray data indicate that the methods are a practical alternative to sparse Gaussian graphical models.

1 Introduction

One way to explore the structure of a high dimensional distribution P for a random vector $X = (X_1, \ldots, X_d)$ is to estimate its undirected graph. The undirected graph G associated with P has d vertices corresponding to the variables $X_1, \ldots, X_d$, and omits an edge between two nodes X_i and X_j if and only if X_i and X_j are conditionally independent given the other variables. Currently, the most popular methods for estimating G assume that the distribution P is Gaussian. Finding the graphical structure in this case amounts to estimating the inverse covariance matrix Ω; the edge between X_j and X_k is missing if and only if $\Omega_{jk} = 0$. Algorithms for optimizing the ℓ_1-regularized log-likelihood have recently been proposed that efficiently produce sparse estimates of the inverse covariance matrix and the underlying graph (Banerjee et al., 2008; Friedman et al., 2007).

In this paper our goal is to relax the Gaussian assumption and to develop nonparametric methods for estimating the graph of a distribution. Of course, estimating a high dimensional distribution is impossible without making any assumptions. The approach we take here is to force the graphical structure to be a 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 family of undirected graphs that are allowed.

If the graph for P is a forest, then its density p can be written as

$$p(x) = \prod_{(i,j)\in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \prod_{k=1}^{d} p(x_k) \tag{1.1}$$

where E is the set of edges in the forest. Thus, it is only necessary to estimate the bivariate and univariate marginals. Given any distribution P with density p, there is a tree T and a density p_T whose graph is T and which is closest in Kullback-Leibler divergence to p. When P is known, then the best fitting tree distribution can be obtained by Kruskal's algorithm (Kruskal, 1956), or other algorithms for finding a maximum weight spanning tree. In the discrete case, the algorithm can be applied to the estimated probability mass function, resulting in a procedure originally proposed by Chow and Liu (1968). Here we are concerned with continuous random variables, and we estimate the bivariate marginals with nonparametric kernel density estimators.

In high dimensions, fitting a fully connected spanning tree can be expected to over fit. We regulate the complexity of the forest by selecting the edges to include using a data splitting scheme, a simple form of

cross validation. In particular, we consider the family of forest structured densities that use the marginal kernel density estimates constructed on the first partition of the data, and estimate the risk of the resulting densities over a second, held out partition. The optimal forest in terms of the held out risk is then obtained by finding a maximum weight spanning forest for an appropriate set of edge weights.

While tree and forest structured density estimation has been long recognized as a useful tool, there has been little theoretical analysis of the statistical properties of such density estimators. The main contribution of this paper is an analysis of the asymptotic properties of forest density estimation in high dimensions. We allow both the sample size n and dimension d to increase, and prove oracle results on the risk of the method. In particular, we assume that the univariate and bivariate marginal densities lie in a Hölder class with exponent β (see Section 4 for details), and show that

$$R(\widehat{p}_{\widehat{F}}) - \min_{F} R(\widehat{p}_F) = O_P\left(\sqrt{\log(nd)}\left(\frac{k^* + \widehat{k}}{n^{\beta/(2+2\beta)}} + \frac{d}{n^{\beta/(1+2\beta)}}\right)\right) \tag{1.2}$$

where R denotes the risk, the expected negative log-likelihood, $\widehat{k}$ is the number of edges in the estimated forest $\widehat{F}$, and k^* is the number of edges in the optimal forest F^* that can be constructed in terms of the kernel density estimates $\widehat{p}$.

Among the only other previous work analyzing tree structured graphical models is Tan et al. (2009a) and Chechetka and Guestrin (2007). Tan et al. (2009a) analyze the error exponent in the rate of decay of the error probability for estimating the tree, in the fixed dimension setting, and Chechetka and Guestrin (2007) give a PAC analysis. An extension to the Gaussian case is given by Tan et al. (2009b).

In addition to the above results on risk consistency, we also study the problem of estimating forests with restricted tree sizes. In many applications, one is interested in obtaining a graphical representation of a high dimensional distribution to aid in interpretation. For instance, a biologist studying gene interaction networks might be interested in a visualization that groups together genes in small sets. Such a clustering approach through density estimation is problematic if the graph is allowed to have cycles, as this can require marginal densities to be estimated with many interacting variables. Restricting the graph to be a forest beats the curse of dimensionality by requiring only univariate and bivariate marginal densities. To group the variables into small interacting sets, we are led to the problem of estimating a maximum weight spanning forest with a restriction on the size of each component tree. As we demonstrate, estimating restricted tree size forests can also be useful in model selection for the purpose of risk minimization. Limiting the tree size gives another way of regulating tree complexity that provides larger family of forest to select from in the data splitting procedure.

While the problem of finding a maximum weight forest with restricted tree size may be natural, it appears not to have been studied previously. We prove that the problem is NP-hard through a reduction from the problem of Exact 3-Cover (Garey & Johnson, 1979), where we are given a set X and a family $\mathcal{S}$ of 3-element subsets of X, and must choose a subfamily of disjoint 3-element subsets to cover X. While finding the exact optimum is hard, we give a practical 4-approximation algorithm for finding the optimal tree restricted forest; that is, our algorithm outputs a forest whose weight is guaranteed to be at least $\frac{1}{4}w(F^*)$, where $w(F^*)$ is the weight of the optimal forest. This approximation guarantee translates into excess risk bounds on the constructed forest using our previous analysis, as the weight of the forest corresponds to contribution to the risk coming from the bivariate marginals over the edges in the forest. Our experimental results with this approximation algorithm show that it can be effective in practice for forest density estimation.

In Section 2 we review some background and notation. In Section 3 we present a two-stage algorithm, and we provide a theoretical analysis of the algorithm in Section 4, with the detailed proofs collected in the full arXiv version of this paper (Liu et al., 2010). In Section 6 we present experiments with both simulated data and gene microarray data, where the problem is to estimate the gene-gene association graph, which has been previously studied using Gaussian graphical models by Wille et al. (2004).

2 Preliminaries and Notation

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 $X_j^{(i)}$ 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 $|E(F)| < d$, noting that we do not restrict the graph to be connected. We say that a probability density function $p(x)$ is *supported by a forest F* if the density can be written as

$$p_F(x) = \prod_{(i,j)\in E(F)} \frac{p(x_i, x_j)}{p(x_i)\, p(x_j)} \prod_{k \in V_F} p(x_k), \tag{2.1}$$

where each $p(x_i, x_j)$ is a bivariate density on $\mathcal{X}_i \times \mathcal{X}_j$, and each $p(x_k)$ is a univariate density on $\mathcal{X}_k$ (Lauritzen, 1996).

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

$$\mathcal{P}_d = \left\{ p \geq 0 : \int_{\mathcal{X}} p(x)\, d\mu(x) = 1, \text{ and } p(x) \text{ satisfies (2.1) for some } F \in \mathcal{F}_d \right\}. \tag{2.2}$$

To bound the number of labeled spanning forests on d nodes, note that each such forest can be obtained by forming a labeled tree on $d+1$ nodes, and then removing node $d+1$. From Cayley's formula (Cayley, 1889; Aigner & Ziegler, 1998), we then obtain the following.

Proposition 2.1 *The size of the collection $\mathcal{F}_d$ of labeled forests on d nodes satisfies*

$$|\mathcal{F}_d| < (d+1)^{d-1}. \tag{2.3}$$

Define the oracle forest density

$$q^* = \arg\min_{q \in \mathcal{P}_d} D(p^* \| \, q) \tag{2.4}$$

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

$$D(p \| \, q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx, \tag{2.5}$$

under the convention that $0 \log(0/q) = 0$, and $p \log(p/0) = \infty$ for $p \neq 0$. The following is proved by Bach and Jordan (2003).

Proposition 2.2 *Let q^* be defined as in* (2.4)*. There exists a tree $T^* \in \mathcal{F}_d$, such that*

$$q^* = p^*_{T^*} = \prod_{(i,j) \in E(T^*)} \frac{p^*(x_i, x_j)}{p^*(x_i)\, p^*(x_j)} \prod_{k \in V_{T^*}} p^*(x_k) \tag{2.6}$$

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

$$R(q) = -\mathbb{E} \log q(X) = -\int_{\mathcal{X}} p^*(x) \log q(x)\, dx. \tag{2.7}$$

It is straightforward to see that the density q^* defined in (2.4) also minimizes the negative log-likelihood loss:

$$q^* = \arg\min_{q \in \mathcal{P}_d} D(p^* \| \, q) = \arg\min_{q \in \mathcal{P}_d} R(q) \tag{2.8}$$

We thus define the oracle risk as $R^* = R(q^*)$. Using Proposition 2.2 and equation (2.1), we have

$$\begin{aligned}
R^* &= R(q^*) = R(p^*_{T^*}) \\
&= -\int_{\mathcal{X}} p^*(x) \Bigg(\sum_{(i,j) \in E(T^*)} \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} + \sum_{k \in V_{T^*}} \log \left(p^*(x_k) \right) \Bigg) dx \\
&= -\sum_{(i,j) \in E(T^*)} \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} dx_i dx_j - \sum_{k \in V_{T^*}} \int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k) dx_k \\
&= -\sum_{(i,j) \in E(T^*)} I(X_i; X_j) + \sum_{k \in V_{T^*}} H(X_k),
\end{aligned} \tag{2.9}$$

where

$$I(X_i; X_j) = \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i)\, p^*(x_j)}\, dx_i dx_j \tag{2.10}$$

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

$$H(X_k) = -\int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k)\, dx_k \tag{2.11}$$

is the entropy. While the best forest will in fact be a spanning tree, the densities $p^*(x_i, x_j)$ are in practice not known. We estimate the marginals using finite data, in terms of a kernel density estimates $\widehat{p}_{n_1}(x_i, x_j)$ over a training set of size n_1. With these estimated marginals, we consider all forest density estimates of the form

$$\widehat{p}_F(x) = \prod_{(i,j)\in E(F)} \frac{\widehat{p}_{n_1}(x_i, x_j)}{\widehat{p}_{n_1}(x_i)\,\widehat{p}_{n_1}(x_j)} \prod_{k\in V_F} \widehat{p}_{n_1}(x_k). \tag{2.12}$$

Within this family, the best density estimate may not be supported on a full spanning tree, since a full tree will in general be subject to over fitting. Analogously, in high dimensional linear regression, the optimal regression model will generally be a full p-dimensional fit, with a nonzero parameter for each variable. However, when estimated on finite data the variance of a full model will dominate the squared bias, resulting in over fitting. In our setting of density estimation we will regulate the complexity of the forest by cross validating over a held out set.

There are several different ways to judge the quality of a forest structured density estimator. In this paper we concern ourselves with prediction and density estimation, and thus focus on risk consistency.

Definition 2.3 ((Risk consistency)) *For an estimator $\widehat{q}_n \in \mathcal{P}_d$, the excess risk is defined as $R(\widehat{q}_n) - R^*$. The estimator $\widehat{q}_n$ is risk consistent with convergence rate δ_n if*

$$\lim_{M\to\infty} \limsup_{n\to\infty} \mathbb{P}\left(R(\widehat{q}_n) - R^* \geq M\delta_n\right) = 0. \tag{2.13}$$

In this case we write $R(\widehat{q}_n) - R^* = O_P(\delta_n)$.

It is important to note that this criterion is an oracle property, in the sense that the true density $p^*(x)$ is not restricted to be supported by a tree; rather, the property assesses how well a given estimator $\widehat{q}$ approximates the best forest density (the oracle) within a class.

3 Kernel Density Estimation For Forests

If the true density $p^*(x)$ were known, by Proposition 2.2, the density estimation problem would be reduced to finding the best tree structure T_d^*, satisfying

$$T_d^* = \operatorname*{arg\,min}_{T\in\mathcal{T}_d} R(p_T^*) = \operatorname*{arg\,min}_{T\in\mathcal{T}_d} D(p^* \,\|\, p_T^*). \tag{3.1}$$

The optimal tree T_d^* can be found by minimizing the right hand side of (2.9). Since the entropy term $H(X) = \sum_k H(X_k)$ is constant across all trees, this can be recast as the problem of finding the maximum weight spanning tree 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 (2.9) by the plug-in estimate $\widehat{I}_n(X_i, X_j)$, defined as

$$\widehat{I}_n(X_i, X_j) = \int_{\mathcal{X}_i\times\mathcal{X}_j} \widehat{p}_n(x_i, x_j) \log \frac{\widehat{p}_n(x_i, x_j)}{\widehat{p}_n(x_i)\,\widehat{p}_n(x_j)}\, dx_i dx_j \tag{3.2}$$

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, we need an automatic data-dependent way to choose it. We adopt the following two-stage procedure. First, randomly partition the data into two sets $\mathcal{D}_1$ and $\mathcal{D}_2$ of sizes n_1 and n_2; 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 (2.1), using the kernel density estimates constructed in Step 1.

Algorithm 3.1 Chow-Liu (Kruskal)

1: **Input** data $\mathcal{D}_1 = \{X^{(1)}, \ldots, X^{(n_1)}\}$.

2: Calculate $\widehat{M}_{n_1}$, according to (3.3), (3.4), and (3.5).

3: Initialize $E^{(0)} = \emptyset$

4: **for** $k = 1, \ldots, d-1$ **do**

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

6: $\quad E^{(k)} \leftarrow E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$

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

3.1 Step 1: Estimating the marginals

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

$$\widehat{p}_{n_1}(x_i, x_j) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_2^2} K\left(\frac{X_i^{(s)} - x_i}{h_2}\right) K\left(\frac{X_j^{(s)} - x_j}{h_2}\right), \tag{3.3}$$

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

$$\widehat{p}_{n_1}(x_k) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_1} K\left(\frac{X_k^{(s)} - x_k}{h_1}\right), \tag{3.4}$$

where $h_1 > 0$ is the univariate bandwidth. Detailed specifications for $K(\cdot)$ and h_1, h_2 will be discussed in the next section.

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

$$\widehat{I}_{n_1}(X_i, X_j) = \frac{1}{m^2} \sum_{k=1}^{m} \sum_{\ell=1}^{m} \widehat{p}_{n_1}(x_{ki}, x_{\ell j}) \log \frac{\widehat{p}_{n_1}(x_{ki}, x_{\ell j})}{\widehat{p}_{n_1}(x_{ki})\, \widehat{p}_{n_1}(x_{\ell j})}. \tag{3.5}$$

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.

3.2 Step 2: Optimizing the forest

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 over fitting in the density estimate. In order to reduce the variance, we prune the tree; that is, we choose forest with $k \leq d-1$ 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 3.1) which corresponds to a forest $\widehat{F}_{n_1}^{(k)}$ with k edges, where $\widehat{F}_{n_1}^{(0)}$ is the union of d disconnected nodes. To select k, we choose among the d trees $\widehat{F}_{n_1}^{(0)}, \widehat{F}_{n_1}^{(1)}, \ldots, \widehat{F}_{n_1}^{(d-1)}$.

Let $\widehat{p}_{n_2}(x_i, x_j)$ and $\widehat{p}_{n_2}(x_k)$ be defined as in (3.3) and (3.4), 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

$$\begin{aligned} &\widehat{R}_{n_2}(p_F) \\ &= -\sum_{(i,j) \in E_F} \int_{\mathcal{X}_i \times \mathcal{X}_j} \widehat{p}_{n_2}(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)\, p(x_j)}\, dx_i dx_j - \sum_{k \in V_F} \int_{\mathcal{X}_k} \widehat{p}_{n_2}(x_k) \log p(x_k)\, dx_k. \end{aligned} \tag{3.6}$$

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

$$\widehat{k} = \underset{k\in\{0,\ldots,d-1\}}{\arg\min}\ \widehat{R}_{n_2}\left(\widehat{p}_{\widehat{F}_{n_1}^{(k)}}\right) \tag{3.7}$$

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

For computational simplicity, we can also estimate $\widehat{k}$ as

$$\widehat{k} = \underset{k\in\{0,\ldots,d-1\}}{\arg\max}\ \frac{1}{n_2}\sum_{s\in\mathcal{D}_2}\log\left(\prod_{(i,j)\in E^{(k)}}\frac{\widehat{p}_{n_1}(X_i^{(s)},X_j^{(s)})}{\widehat{p}_{n_1}(X_i^{(s)})\,\widehat{p}_{n_1}(X_j^{(s)})}\prod_{k\in V_{\widehat{F}_{n_1}^{(k)}}}\widehat{p}_{n_1}(X_k^{(s)})\right) \tag{3.8}$$

$$= \underset{k\in\{0,\ldots,d-1\}}{\arg\max}\ \frac{1}{n_2}\sum_{s\in\mathcal{D}_2}\log\left(\prod_{(i,j)\in E^{(k)}}\frac{\widehat{p}_{n_1}(X_i^{(s)},X_j^{(s)})}{\widehat{p}_{n_1}(X_i^{(s)})\,\widehat{p}_{n_1}(X_j^{(s)})}\right). \tag{3.9}$$

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 density estimate is given by

$$\widehat{p}_n(x) = \prod_{(i,j)\in E^{(\widehat{k})}}\frac{\widehat{p}_{n_1}(x_i,x_j)}{\widehat{p}_{n_1}(x_i)\,\widehat{p}_{n_1}(x_j)}\prod_k\widehat{p}_{n_1}(x_k). \tag{3.10}$$

4 Statistical Properties

In this section we present our theoretical results on risk consistency and structure selection consistency of the forest density estimate $\widehat{p}_n = \widehat{p}_{\widehat{F}_d^{(\widehat{k})}}$.

To establish some notation, we write $a_n = \Omega(b_n)$ if there exists a constant c such that $a_n \geq c b_n$ for sufficiently large n. We also write $a_n \asymp b_n$ if there exists a constant c such that $a_n \leq c\, b_n$ and $b_n \leq c\, a_n$ for sufficiently large n. Given a d-dimensional function f on the domain $\mathcal{X}$, we denote its $L_2(P)$-norm and sup-norm as

$$\|f\|_{L_2(P)} = \sqrt{\int_{\mathcal{X}} f^2(x)dP_X(x)}, \qquad \|f\|_\infty = \sup_{x\in\mathcal{X}}|f(x)| \tag{4.1}$$

where P_X is the probability measure induced by X. Throughout this section, all constants are treated as generic values, and as a result they can change from line to line.

In our use of a data splitting scheme, we always adopt equally sized splits for simplicity, so that $n_1 = n_2 = n/2$, noting that this does not affect the final rate of convergence.

4.1 Assumptions on the density

Fix $\beta > 0$. For any d-tuple $\alpha = (\alpha_1,\ldots,\alpha_d) \in \mathbb{N}^d$ and $x = (x_1,\ldots,x_d) \in \mathcal{X}$, we define $x^\alpha = \prod_{j=1}^d x_j^{\alpha_j}$. Let D^α denote the differential operator

$$D^\alpha = \frac{\partial^{\alpha_1+\cdots+\alpha_d}}{\partial x_1^{\alpha_1}\cdots\partial x_d^{\alpha_d}}. \tag{4.2}$$

For any real-valued d-dimensional function f on $\mathcal{X}$ that is $\lfloor\beta\rfloor$-times continuously differentiable at point $x_0 \in \mathcal{X}$, let $P_{f,x_0}^{(\beta)}(x)$ be its Taylor polynomial of degree $\lfloor\beta\rfloor$ at point x_0:

$$P_{f,x_0}^{(\beta)}(x) = \sum_{\alpha_1+\cdots+\alpha_d\leq\lfloor\beta\rfloor}\frac{(x-x_0)^\alpha}{\alpha_1!\cdots\alpha_d!}D^\alpha f(x_0). \tag{4.3}$$

Fix $L > 0$, and denote by $\Sigma(\beta,L,r,x_0)$ the set of functions $f : \mathcal{X} \to \mathbb{R}$ that are $\lfloor\beta\rfloor$-times continuously differentiable at x_0 and satisfy

$$\left|f(x) - P_{f,x_0}^{(\beta)}(x)\right| \leq L\|x-x_0\|_2^\beta, \ \ \forall x \in \mathcal{B}(x_0,r) \tag{4.4}$$

where $\mathcal{B}(x_0,r) = \{x : \|x-x_0\|_2 \leq r\}$ is the L_2-ball of radius r centered at x_0. The set $\Sigma(\beta,L,r,x_0)$ is called the (β,L,r,x_0)-locally Hölder class of functions. Given a set A, we define

$$\Sigma(\beta,L,r,A) = \cap_{x_0\in A}\Sigma(\beta,L,r,x_0). \tag{4.5}$$

The following are the regularity assumptions we make on the true density function $p^*(x)$.

Assumption 4.1 *For any $1 \leq i < j \leq d$, we assume*

(D1) there exist $L_1 > 0$ and $L_2 > 0$ such that for any $c > 0$ the true bivariate and univariate densities satisfy

$$p^*(x_i, x_j) \in \Sigma\left(\beta, L_2, c\,(\log n/n)^{\frac{1}{2\beta+2}}, \mathcal{X}_i \times \mathcal{X}_j\right) \tag{4.6}$$

and

$$p^*(x_i) \in \Sigma\left(\beta, L_1, c\,(\log n/n)^{\frac{1}{2\beta+1}}, \mathcal{X}_i\right); \tag{4.7}$$

(D2) there exists two constants c_1 and c_2 such that

$$c_1\gamma_n \leq \inf_{x_i,x_j \in \mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \leq \sup_{x_i,x_j \in \mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \leq c_2 \tag{4.8}$$

μ-almost surely, where $\gamma_n^2 = \Omega\left(\sqrt{\dfrac{\log n + \log d}{n^{\beta/(\beta+1)}}}\right)$.

These assumptions are mild, in the sense that instead of adding constraints on the joint density $p^*(x)$, we only add regularity conditions on the bivariate and univariate marginals.

4.2 Assumptions on the kernel

An important ingredient in our analysis is an exponential concentration result for the kernel density estimate, due to Giné and Guillou (2002). We first specify the requirements on the kernel function $K(\cdot)$.

Let $(\Omega, \mathcal{A})$ be a measurable space and let $\mathcal{F}$ be a uniformly bounded collection of measurable functions.

Definition 4.2 *$\mathcal{F}$ is a bounded measurable VC class of functions with characteristics A and v if it is separable and for every probability measure P on $(\Omega, \mathcal{A})$ and any $0 < \epsilon < 1$,*

$$N\left(\epsilon \|F\|_{L_2(P)}, \mathcal{F}, \|\cdot\|_{L_2(P)}\right) \leq \left(\frac{A}{\epsilon}\right)^v, \tag{4.9}$$

where $F(x) = \sup_{f\in\mathcal{F}} |f(x)|$ and $N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(P)})$ denotes the ϵ-covering number of the metric space $(\Omega, \|\cdot\|_{L_2(P)})$; that is, the smallest number of balls of radius no larger than ϵ (in the norm $\|\cdot\|_{L_2(P)}$) needed to cover $\mathcal{F}$.

The one-dimensional density estimates are constructed using a kernel K, and the two-dimensional estimates are constructed using the product kernel

$$K_2(x, y) = K(x) \cdot K(y). \tag{4.10}$$

Assumption 4.3 *The kernel K satisfies the following properties.*

(K1) $\displaystyle\int K(u)\,du = 1, \int_{-\infty}^{\infty} K^2(u)\,du < \infty$ *and* $\displaystyle\sup_{u\in\mathbb{R}} K(u) \leq c$ *for some constant c.*

(K2) K is a finite linear combination of functions g whose epigraphs epi$(g) = \{(s, u) : g(s) \geq u\}$, can be represented as a finite number of Boolean operations (union and intersection) among sets of the form $\{(s, u) : Q(s, u) \geq \phi(u)\}$, where Q is a polynomial on $\mathbb{R} \times \mathbb{R}$ and ϕ is an arbitrary real function.

(K3) K has a compact support and for any $\ell \geq 1$ and $1 \leq \ell' \leq \lfloor\beta\rfloor$

$$\int |t|^\beta\,|K(t)|\,dt < \infty, \text{ and } \int |K(t)|^\ell dt < \infty, \quad \int t^{\ell'} K(t)dt = 0. \tag{4.11}$$

Assumptions (K1), (K2) and (K3) are mild. As pointed out by Nolan and Pollard (1987), both the pyramid (truncated or not) kernel and the boxcar kernel satisfy them. It follows from (K2) that the classes of functions

$$\mathcal{F}_1 = \left\{\frac{1}{h_1}K\left(\frac{u-\cdot}{h_1}\right) : u \in \mathbb{R},\ h_1 > 0\right\} \tag{4.12}$$

$$\mathcal{F}_2 = \left\{\frac{1}{h_2^2}K\left(\frac{u-\cdot}{h_2}\right)K\left(\frac{t-\cdot}{h_2}\right) : u, t \in \mathbb{R},\ h_2 > 0\right\} \tag{4.13}$$

are bounded VC classes, in the sense of Definition 4.2. Assumption (K3) essentially says that the kernel $K(\cdot)$ should be *β-valid*; see Tsybakov (2008) and Definition 6.1 in Rigollet and Vert (2009) for further details about this assumption.

We choose the bandwidths h_1 and h_2 used in the one-dimensional and two-dimensional kernel density estimates to satisfy

$$h_1 \asymp \left(\frac{\log n}{n}\right)^{\frac{1}{1+2\beta}} \tag{4.14}$$

$$h_2 \asymp \left(\frac{\log n}{n}\right)^{\frac{1}{2+2\beta}}. \tag{4.15}$$

This choice of bandwidths ensures the optimal rate of convergence.

4.3 Risk consistency

Given the above assumptions, we first present a key lemma that establishes the rates of convergence of bivariate and univariate kernel density estimates in the $\sup$ norm. Due to space limitations, the proof of this and our other technical results are provided in the extended arXiv version of this paper (Liu et al., 2010).

Lemma 4.4 *Under Assumptions 4.1 and 4.3, and choosing bandwidths satisfying* (4.14) *and* (4.15)*, the bivariate and univariate kernel density estimates $\widehat{p}(x_i, x_j)$ and $\widehat{p}(x_k)$ in* (3.3) *and* (3.4) *satisfy*

$$\max_{(i,j)\in\{1,\ldots,d\}\times\{1,\ldots,d\}} \sup_{(x_i,x_j)\in\mathcal{X}_i\times\mathcal{X}_j} |\widehat{p}(x_i,x_j) - p^*(x_i,x_j)| = O_P\left(\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}}\right) \tag{4.16}$$

and

$$\max_{k\in\{1,\ldots,d\}} \sup_{x_k\in\mathcal{X}_k} |\widehat{p}(x_k) - p^*(x_k)| = O_P\left(\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}\right). \tag{4.17}$$

To describe the risk consistency result, let $\mathcal{P}_d^{(d-1)} = \mathcal{P}_d$ be the family of densities that are supported by forests with at most $d-1$ edges, as already defined in (2.2). For $0 \le k \le d-1$, we define $\mathcal{P}_d^{(k)}$ as the family of d-dimensional densities that are supported by forests with at most k edges. Then

$$\mathcal{P}_d^{(0)} \subset \mathcal{P}_d^{(1)} \subset \cdots \subset \mathcal{P}_d^{(d-1)}. \tag{4.18}$$

Now, due to the nesting property (4.18), we have

$$\inf_{q_F\in\mathcal{P}_d^{(0)}} R(q_F) \ge \inf_{q_F\in\mathcal{P}_d^{(1)}} R(q_F) \ge \cdots \ge \inf_{q_F\in\mathcal{P}_d^{(d-1)}} R(q_F). \tag{4.19}$$

We first analyze the forest density estimator obtained using a fixed number of edges $k < d$; specifically, consider stopping the Chow-Liu algorithm in Stage 1 after k iterations. This is in contrast to the algorithm described in 3.2, where the pruned tree size is automatically determined on the held out data. While this is not very realistic in applications, since the tuning parameter k is generally hard to choose, the analysis in this case is simpler, and can be directly exploited to analyze the more complicated data-dependent method.

Theorem 4.5 (Risk consistency) *Let $\widehat{p}_{\widehat{F}_d^{(k)}}$ be the forest density estimate with $|E(\widehat{F}_d^{(k)})| = k$, obtained after the first k iterations of the Chow-Liu algorithm, for some $k \in \{0,\ldots,d-1\}$. Under Assumptions 4.1 and 4.3, we have*

$$R(\widehat{p}_{\widehat{F}_d^{(k)}}) - \inf_{q_F\in\mathcal{P}_d^{(k)}} R(q_F) = O_P\left(k\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}\right). \tag{4.20}$$

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.

The above results can be used to prove a risk consistency result for the data-dependent pruning method using the data-splitting scheme described in Section 3.2.

Theorem 4.6 *Let $\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}$ be the forest density estimate using the data-dependent pruning method in Section 3.2, and 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. Under Assumptions 4.1 and 4.3, we have*

$$R(\widehat{p}_{\widehat{F}_d^{(\widehat{k})}}) - \min_{0\le k\le d-1} R(\widehat{p}_{\widehat{F}_d^{(k)}}) = O_P\left((k^* + \widehat{k})\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d\sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}\right) \tag{4.21}$$

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

The proof of this theorem is given in (Liu et al., 2010).

Algorithm 5.1 Approximate Max Weight t-Restricted Forest

1: **Input** graph G with positive edge weights, and positive integer $t \geq 2$.

2: Sort edges in decreasing order of weight.

3: Greedily add edges in decreasing order of weight such that
   (a) the degree of any node is at most t;
   (b) no cycles are formed.
   The resulting forest is $F' = \{T_1, T_2, \ldots, T_m\}$.

4: **Output** $F_t = \cup_j \texttt{TreePartition}(T_j, t)$.

5 Tree Restricted Forests

We now turn to the problem of estimating forests with restricted tree sizes. As discussed in the introduction, clustering problems motivate the goal of constructing forest structured density estimators where each connected component has a restricted number of edges. But estimating restricted tree size forests can also be useful in model selection for the purpose of risk minimization, since the maximum subtree size can be viewed as an additional complexity parameter.

Definition 5.1 *A t-restricted forest of a graph G is a subgraph F_t such that*

1. *F_t is the disjoint union of connected components $\{T_1, ..., T_m\}$, each of which is a tree;*
2. *$|T_i| \leq t$ for each $i \leq m$, where $|T_i|$ denotes the number of edges in the ith component.*

Given a weight w_e assigned to each edge of G, an optimal t-restricted forest F_t^* satisfies

$$w(F_t^*) \geq \max_{\mathcal{F}_t(G)} w(F_t) \tag{5.1}$$

where $w(F) = \sum_{e \in F} w_e$ is the weight of a forest F and $\mathcal{F}_t(G)$ denotes the collection of all t-restricted forests of G.

For $t = 1$, the problem is maximum weighted matching. Unfortunately for $t \geq 2$, determining a maximum weight t-restricted forest is an NP-hard problem; however, this problem appears not to have been previously studied. Our reduction is from Exact 3-Cover (X3C), shown to be NP-complete by Garey and Johnson (1979)). In X3C, we are given a set X, a family $\mathcal{S}$ of 3-element subsets of X, and we must choose a subfamily of disjoint 3-element subsets to cover X.

Our reduction constructs a graph with special tree-shaped subgraphs called *gadgets*, such that each gadget corresponds to a 3-element subset in $\mathcal{S}$. We show that finding a maximum weight t-restricted forest on this graph would allow us to then recover a solution to X3C by analyzing how the optimal forest must partition each of the gadgets.

Given the difficulty of finding an optimal t-restricted forest, it is of interest to study approximation algorithms. Algorithm 5.1 gives a procedure that has two stages. In the first stage, a forest is greedily constructed in such a way that each node has degree no larger than $t + 1$. In the second stage, each tree in the forest is partitioned in an optimal way by removing edges, resulting in a collection of trees, each of which has size at most t. The second stage employs a procedure we call `TreePartition` that takes a tree and returns the optimal t-restricted subforest. `TreePartition` is a divide-and-conquer procedure of Lukes (1974) that finds a carefully chosen set of forest partitions for each child subtree. It then merges these sets with the parent node one subtree at a time. The details of the `TreePartition` procedure are given in (Liu et al., 2010).

Theorem 5.2 *Let F_t be the output of Algorithm 5.1, and let F_t^* be the optimal t-restricted forest. Then $w(F_t) \geq \frac{1}{4} w(F_t^*)$.*

5.1 Pruning Based on t-Restricted Forests

For a given t, after producing an approximate maximum weight t-restricted forest $\widehat{F}_t$ using $\mathcal{D}_1$, we prune away edges using $\mathcal{D}_2$. To do so, we first construct a new set of univariate and bivariate kernel density estimates using $\mathcal{D}_2$, as before, $\widehat{p}_{n_2}(x_i)$ and $\widehat{p}_{n_2}(x_i, x_j)$. We then estimate the "cross-entropies" of the kernel density estimates $\widehat{p}_{n_1}$ for each pair of variables by computing

$$\widehat{I}_{n_2,n_1}(X_i, X_j) = \int \widehat{p}_{n_2}(x_i, x_j) \log \frac{\widehat{p}_{n_1}(x_i, x_j)}{\widehat{p}_{n_1}(x_i)\widehat{p}_{n_1}(x_j)} \, dx_i \, dx_j \tag{5.2}$$

$$\approx \frac{1}{m^2} \sum_{k=1}^{m} \sum_{\ell=1}^{m} \widehat{p}_{n_2}(x_{ki}, x_{\ell j}) \log \frac{\widehat{p}_{n_1}(x_{ki}, x_{\ell j})}{\widehat{p}_{n_1}(x_{ki}) \, \widehat{p}_{n_1}(x_{\ell j})}. \tag{5.3}$$

Algorithm 5.2 t-Restricted Forest Density Estimation

1: Divide data into two halves $\mathcal{D}_1$ and $\mathcal{D}_2$.

2: Compute kernel density estimators $\widehat{p}_{n_1}$ and $\widehat{p}_{n2}$ for all pairs and single variable marginals.

3: For all pairs (i, j) compute $\widehat{I}_{n_1}(X_i, X_j)$ according to (3.5) and $\widehat{I}_{n_2,n_1}(X_i, X_j)$ according to (5.3).

4: For $t = 0, \ldots, t_{\text{final}}$ where t_{final} is chosen based on the application

   1. Compute or approximate (for $t \geq 2$) the optimal t-restricted forest $\widehat{F}_t$ using $\widehat{I}_{n_1}$ as edge weights.
   2. Prune $\widehat{F}_t$ to eliminate all edges with negative weights $\widehat{I}_{n_2,n_1}$.

5: Among all pruned forests $\widehat{p}_{F^t}$, select $\widehat{t} = \arg\min_{0 \leq t \leq t_{\text{final}}} \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_t})$.

We then eliminate all edges (i, j) in $\widehat{F}_t$ for which $\widehat{I}_{n_2,n_1}(X_i, X_j) \leq 0$. For notational simplicity, we denote the resulting pruned forest again by $\widehat{F}_t$.

To estimate the risk, we simply use $\widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_t})$ as defined before, and select the forest $\widehat{F}_{\widehat{t}}$ according to

$$\widehat{t} = \underset{0 \leq t \leq d-1}{\arg\min}\ \widehat{R}_{n_2}(\widehat{p}_{\widehat{F}_t}). \tag{5.4}$$

The resulting procedure is summarized in Algorithm 5.2.

Using the approximation guarantee and our previous analysis, we have that the population weights of the approximate t-restricted forest and the optimal forest satisfy the following inequality. We state the result for a general c-approximation algorithm; for the algorithm given above, $c = 4$, but tighter approximations are possible.

Theorem 5.3 *Assume the conditions of Theorem 4.5. For $t \geq 2$, let $\widehat{F}_t$ be the forest constructed using a c-approximation algorithm, and let F_t^* be the optimal forest; both constructed with respect to finite sample edge weights $\widehat{w}_{n_1} = \widehat{I}_{n_1}$. Then*

$$w(\widehat{F}_t) \geq \frac{1}{c} w(F_t^*) + O_P\left((k^* + \widehat{k})\sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}}\right) \tag{5.5}$$

where $\widehat{k}$ and k^* are the number of edges in $\widehat{F}_t$ and F_t^*, respectively, and w denotes the population weights, given by the mutual information.

As seen below, although the approximation algorithm has weaker theoretical guarantees, it out-performs other approaches in experiments.

6 Experimental Results

In this section, we report numerical results on both synthetic datasets and microarray data; additional experiments and further details are presented in the extended version of this paper (Liu et al., 2010). We mainly compare the forest density estimator with sparse Gaussian graphical models, fitting a multivariate Gaussian with a sparse inverse covariance matrix. The sparse Gaussian models are estimated using the graphical lasso algorithm (glasso) of Friedman et al. (2007), which is a refined version of an algorithm first derived by Banerjee et al. (2008). Since the glasso typically results in a large parameter bias as a consequence of the ℓ_1 regularization, we also compare with a method that we call the *refit glasso*, which is a two-step procedure—in the first step, a sparse inverse covariance matrix is obtained by the glasso; in the second step, a Gaussian model is refit without ℓ_1 regularization, but enforcing the sparsity pattern obtained in the first step.

6.1 Synthetic data

We generate high dimensional Gaussian and non-Gaussian data which are consistent with an undirected graph. A typical run showing the held-out log-likelihood and estimated graphs is provided in Figure 6.1. We see that for the Gaussian data, the refit glasso has a higher held-out log-likelihood than the forest density estimator and the glasso. This is expected, since the Gaussian model is correct. For very sparse models, however, the performance of the glasso is worse than that of the forest density estimator, due to the large parameter bias resulting from the ℓ_1 regularization. We also observe an efficiency loss in the nonparametric forest density estimator, compared to the refit glasso. The graphs are automatically selected using the held-out log-likelihood, and we see that the nonparametric forest-based kernel density estimator tends to select a sparser model, while the parametric Gaussian models tend to overselect.

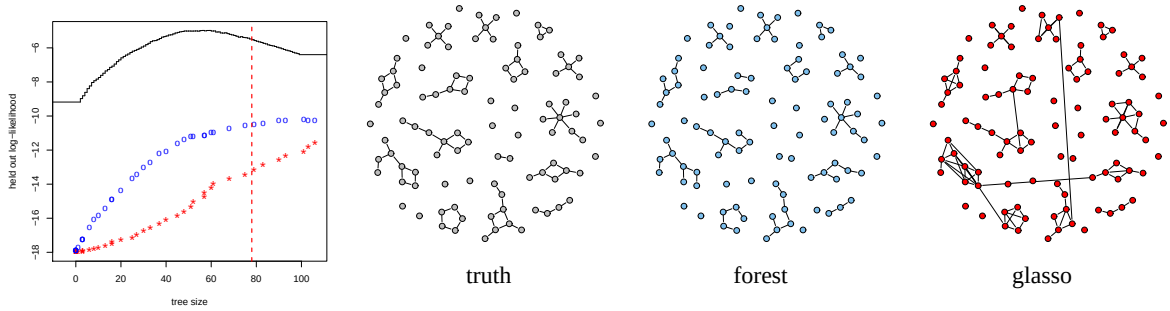

Figure 6.1: Synthetic data, non-Gaussian. Held-out log-likelihood plots show forest density (black step function), glasso (red stars), and refit glasso (blue circles); vertical indicates size of true graph.

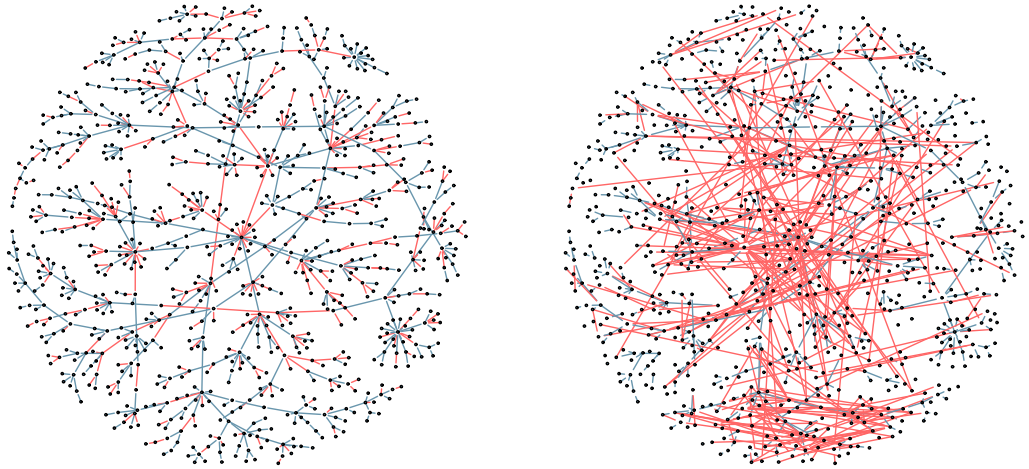

Figure 6.2: A 934 gene subgraph of the full estimated 4238 gene network. Left: estimated forest graph. Right: estimated Gaussian graph. Red edges in the forest graph are missing from the Gaussian graph and vice versa; the blue edges are shared by both graphs. Note that the layout of the genes is the same for both graphs.

6.2 Microarray Data

Our data comes from Nayak et al. (2009). The dataset contains Affymetrics chip measured expression levels of 4238 genes for 295 normal subjects in the *Centre d'Etude du Polymorphisme Humain* (CEPH) and the International HapMap collections. The 295 subjects come from four different groups: 148 unrelated grandparents in the CEPH-Utah pedigrees, 43 Han Chinese in Beijing, 44 Japanese in Tokyo, and 60 Yoruba in Ibadan, Nigeria. Since we want to find common network patterns across different groups of subjects, we pooled the data together into a $n = 295$ by $p = 4238$ numerical matrix.

We estimate the full 4238 node graph using both the forest density estimator (described in Section 3.1 and 3.2) and the Meinshausen-Bühlmann neighborhood search method (Meinshausen & Bühlmann, 2006) with regularization parameter chosen to give it about same number as edges as the forest graph. The forest density estimated graph reveals one strongly connected component of more than 3000 genes and various isolated genes; this is consistent with the analysis in Nayak et al. (2009) and is realistic for the regulatory system of humans. The Gaussian graph contains similar component structure, but the set of edges differs significantly. We also ran the t-restricted forest algorithm for $t = 2000$ and it successfully separates the giant component into three smaller components. Since the forest density estimator produces a sparse and interpretable graph whose structure is consistent with biological analysis, we believe that it may be helpful for studying gene interaction networks.

For visualization purposes, we show only a 934 gene subgraph of the strongly connected component among the full 4238 node graphs we estimated. We refer the reader to the extended arXiv version of this paper (Liu et al., 2010) for the full graph and other visualizations.

7 Conclusion

We have studied forest density estimation for high dimensional data. Forest density estimation skirts the curse of dimensionality by restricting to undirected graphs without cycles, while allowing fully nonparametric marginal densities. The method is computationally simple, and the optimal size of the forest can be robustly selected by a data-splitting scheme. We have established oracle properties and rates of convergence for function estimation in this setting. Our experimental results compared the forest density estimator to the sparse Gaussian graphical model in terms of both predictive risk and the qualitative properties of the estimated graphs for human gene expression array data. Together, these results indicate that forest density estimation can be a useful tool for relaxing the normality assumption in graphical modeling.

Acknowledgements

The research reported here was supported in part by NSF grant CCF-0625879, AFOSR contract FA9550-09-1-0373, and a grant from Google. We thank Haijie Gu for assistance in running the human gene expression data experiments.

References

Aigner, M., & Ziegler, G. (1998). *Proofs from THE BOOK*. Springer-Verlag.

Bach, F. R., & Jordan, M. I. (2003). Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, *4*, 1205–1233.

Banerjee, O., El Ghaoui, L., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, *9*, 485–516.

Cayley, A. (1889). A theorem on trees. *Quart. J. Math.*, *23*, 376–378.

Chechetka, A., & Guestrin, C. (2007). Efficient principled learning of thin junction trees. *In Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada.

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462–467.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*, 432–441.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of np-completeness*. W. H. Freeman.

Giné, E., & Guillou, A. (2002). Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'institut Henri Poincaré (B), Probabilités et Statistiques*, *38*, 907–921.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, *7*, 48–50.

Lauritzen, S. L. (1996). *Graphical models*. Clarendon Press.

Liu, H., Xu, M., Gu, H., Gupta, A., Lafferty, J., & Wasserman, L. (2010). Forest density estimation. arXiv:1001.1557.

Lukes, J. A. (1974). Efficient algorithm for the partitioning of trees. *IBM Jour. of Res. and Dev.*, *18*, 274.

Meinshausen, N., & Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, *34*, 1436–1462.

Nayak, R., Kearns, M., Spielman, R., & Cheung, V. (2009). Coexpression network based on natural variation in human gene expression reveals gene interactions and functions. *Genome Research*, *19*, 1953–1962.

Nolan, D., & Pollard, D. (1987). U-processes: Rates of convergence. *The Annals of Statistics*, *15*, 780 – 799.

Rigollet, P., & Vert, R. (2009). Fast rates for plug-in estimators of density level sets. *Bernoulli (to appear)*.

Tan, V., Anandkumar, A., Tong, L., & Willsky, A. (2009a). A large-deviation analysis for the maximum likelihood learning of tree structures. arXiv:0905.0940.

Tan, V., Anandkumar, A., & Willsky, A. (2009b). Learning Gaussian tree models: Analysis of error exponents and extremal structures. arXiv:0909.5216.

Tsybakov, A. B. (2008). *Introduction to nonparametric estimation*. Springer Publishing Company, Incorporated.

Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W., & Bühlmann, P. (2004). Sparse Gaussian graphical modelling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biology*, *5*, R92.