The Pennsylvania State University
The Graduate School

AUGMENTING THE BOOTSTRAP TO ANALYZE HIGH-DIMENSIONAL GENOMIC DATA

A Dissertation in
Statistics
by
Svitlana Tyekucheva

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2008

The dissertation of Svitlana Tyekucheva was reviewed and approved* by the following:

Francesca Chiaromonte
Associate Professor of Statistics and Health Evaluation Sciences
Dissertation Advisor, Chair of Committee

Bing Li
Professor of Statistics

David Hunter
Associate Professor of Statistics

Kateryna Makova
Associate Professor of Biology

Bruce G. Lindsay
Willaman Professor of Statistics
Head of the Department of Statistics

*Signatures are on file in the Graduate School.

Abstract

The data produced by contemporary high-throughput genomic techniques are often high dimensional and undersampled. In these settings, several statistical analyses become problematic. Among these are techniques that require the inversion of variance-covariance matrices, such as those pursuing supervised dimension reduction or the assessment of interdependence structures, and classification and regression techniques prone to overfitting. In this thesis we show how the ideas of bagging and smoothed bootstrap can be used to overcome undersampling and improve the performance of a number of statistical procedures widely used in genomic applications. We investigate the conditions under which our method, which we call *augmented bootstrap*, improves estimation and demonstrate its performance on simulated data and on data derived from genomic DNA sequences and microarray experiments.

Table of Contents

List of Figures **vi**

List of Tables **viii**

Acknowledgments **x**

Chapter 1
Resampling methods and challenges in the analysis of genomic data **1**
1.1 Introduction and Background 1
1.2 A Motivating Application 6
1.3 The Augmented Bootstrap 8
1.4 Overview of the following chapters 10

Chapter 2
Some analytical results concerning the augmented bootstrap **12**
2.1 Preliminaries 12
2.1.1 Bagging 13
2.1.2 Smoothing 16
2.2 The Augmented Bootstrap 20
2.3 Concluding remarks 23

Chapter 3
Augmented bootstrap for inverse variance-covariance matrices **25**
3.1 Introduction . . . 25
3.2 Simulations . . . 29
3.3 Applications to genomic data . . . 37
3.3.1 DNA sequence data: supervised dimension reduction and classification based on motif frequencies . . . 37
3.3.2 Microarray data: Gene interactions and networks based on transcription profiles . . . 39
3.4 Discussion and modifications . . . 43
3.4.1 Why *simulating* a sphere? . . . 43
3.4.2 ... And why a *sphere*? . . . 46
3.5 Computational burden . . . 49

Chapter 4
Augmented bootstrap for classification and regression trees **51**
4.1 Introduction . . . 51
4.2 Simulations . . . 53
4.3 Application to genomic data . . . 55

Chapter 5
Conclusions and future work **61**

Appendix A
Mean squared error calculations **65**
A.1 Bagging . . . 66
A.2 Smoothing . . . 67
A.3 Augmented bootstrap . . . 70

Appendix B
Supplementary tables **73**

Appendix C
Selected R code **76**
C.1 Inverse variance-covariance matrix simulations . . . 76
C.2 CART checkerboard simulations . . . 79

Bibliography **85**

List of Figures

3.1 Plots of the mean RSE for AB against the sample size multiplier m (with τ^2 fixed at its optimal values; panel A), against the smoothing variance τ^2 (with $m = 70$; panel B). The data were generated from a Gaussian distribution using Σ_1 and $\rho = 0.5$. Different lines correspond to different sample sizes: *solid* for $n_1 = 10p = 1000$, *dotted* for $n_2 = 1.1p = 110$, and *dashed* for $n_3 = 0.5p = 50$. Corresponding to these sample sizes, the values of mean RSE for T are 0.151, 2417.72 and 0.994, respectively. 35

3.2 Panel A: smoothed distributions of partial correlation coefficients obtained with the AB (*solid* line), SH (*dashed* line) and BB (*dotted* line) methods. Panel B: scatterplot of partial correlation coefficients obtained with AB against those obtained with SH. The 1:1 diagonal is superimposed for visualization purposes. Gene pairs with extreme AB partial correlations but moderate SH partial correlations are shown with solid black circles. 42

3.3 Smoothed degree distributions for the networks resulting from the top 2% partial correlations obtained with the AB (*solid* line), SH (*dashed* line) and BB (*dotted* line) methods. 44

4.1 AB CART mean success rates for the chekerboard simulated data against the sample size multiplier m (with optimal $\tau^2 = 0.15$; panel A), and the smoothing variance τ^2 (with $m = 20$; panel B). Points represent mean success rates and a lowess smooth is superimposed for visualization. The darkened point shows the maximum success rate achieved with $m = 20$ and $\tau^2 = 0.15$ 54

4.2 Smooth histograms of class probabilities at a leaf node for the chekerboard simulation data (tree topology fixed). Panel A shows probabilities adjusted with shrinking formulae for a range of values of λ (the shrinkage intensity), and those obtained with noising for a range of values of τ^2 (the smoothing variance). Panel B shows representative smooth histograms (shrinkage methods with $\lambda = 0.5$, and noising with $\tau^2 = 0.5$). Black corresponds to the non-modified class probabilities in both panels. 56
4.3 AB CART mean success rates for the substitution rates classification problem against the sample size multiplier m (with optimal $\tau^2 = 0.5$; panel A), and the smoothing variance τ^2 (with $m = 95$; panel B). Points represent mean success rates and a lowess smooth is superimposed for visualization. The darkened point shows the maximum success rate achieved with $m = 95$ and $\tau^2 = 0.5$ 59

List of Tables

1.1 Leave-one-out cross validation success rates for the X inactivation problem. Es(cape) genes are genes that escape inactivation. Su(bject) genes are inactivated genes. 100, 200 and 500 Kb are sizes of windows around the genes' transcription start sites. 8

3.1 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.5$. 33

3.2 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n. Non-Gaussian simulation data; $p = 100$ and $\rho = 0.5$. 34

3.3 Optimal smoothing variance τ^2 of the AB method for various Gaussian (columns 1–3) and non-Gaussian (column 4) simulation scenarios, and choices of n; $p = 100$ and $\rho = 0.5$. 36

3.4 Leave-one-out cross-validation success rates for various methods, X-inactivation data. 39

3.5 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n. Gaussian simulation data; $p = 100$ and Σ_1, with $\rho = 0.5$. 46

3.6 Mean (and standard deviation) of the Relative Squared Error for adaptive augmented bootstrap, outer bagged regularized, and modified bagged regularized estimators of Σ^{-1} with the adaptive target. Various choices of n. Gaussian simulation data; $p = 100$ and Σ_1, with $\rho = 0.5$. 47

4.1 Mean (and standard deviation) of the success rates for various tree-based classifiers. Checkerboard simulation data with $p = 2$, $n = 15$ training points, and $k = 200$ test points. 54
4.2 Leave-one-out cross-validation success rates for the substitution rates classification problem. For traditional CART and shrinkage with optimal value of the tuning parameter only one success rate is reported. For bagging and augmented bootstrap with optimal values of the tuning parameters mean and standard deviation (parenthetically) of success rates over 50 replications of the resampling procedures are reported. 59

B.1 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.1$. 73
B.2 Optimal smoothing variance τ^2 for various choices of Σ, n, $p = 100$ and $\rho = 0.1$. 74
B.3 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.9$. 74
B.4 Optimal smoothing variance τ^2 for various choices of Σ, n, $p = 100$ and $\rho = 0.9$. 74
B.5 Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n; $p = 100$, with $\rho = 0.5$; Σ_2, Σ_3, $\tilde{\Sigma}$ - as described in Section 3.2 75

Acknowledgments

I am grateful to the members of my dissertation committee Francesca Chiaromonte, Bing Li, David Hunter, and Kateryna Makova for sharing their time and wisdom to improve my work. My dissertation advisor Francesca was an incredible mentor and I would not have been able make it this far without her invaluable support. Kateryna's expertise in biology greatly influenced my interest in genomic applications. Bing and David were wonderful teachers from whom I learned lot.

I am also thankful to Webb Miller for his encouragement, advice and support throughout my years at the Center for Comparative Genomics and Bioinformatics. My gratitude also goes to Ross Hardison for his advice throughout the research process.

Finally, none of this would be possible without a very supportive network of friends and family.

CHAPTER 1

Resampling methods and challenges in the analysis of genomic data

1.1 Introduction and Background

Today many applications involve analysis of data with limited amount of observations in a high dimensional space. In genomics applications, in which we are interested the most, such setup occurs very often. Contemporary experimental and bioinformatic techniques often allow us to measure many features, but usually at a limited number of loci, and/or with a limited number of replicates. Features can be both continuous and discrete/categorical. Many questions of biological interest can be addressed by supervised methods, such as regression or classification analyses in which a response Y (quantitative or categorical) is studied as a function of a large predictor vector $X \in \mathbb{R}^p$. In these regression and classification analyses, sufficient dimension reduction approaches (Cook, 1998) are very popular. Sufficient dimension reduction methods concern estimating directions (linear combinations of predictors) that span the so-called central subspace. These linear

combinations contain all relevant information about the distribution of $Y|X$. Traditional methods to estimate directions in the central subspace include ordinary least squares, which estimates only one direction and applies when Y is continuous, and sliced inverse regression (Li, 1991, SIR), which can estimate several directions and can be applied both for Y continuous (in this case Y is "sliced" into several levels) and Y categorical. Estimating directions in the central subspace by the aforementioned traditional methods is problematic since it involves inversion of the sample variance-covariance matrix of the predictors, which is singular when we are given only n distinct observations in dimension $p > n$. Even when $p < n$ and the sample variance-covariance matrix of the predictors is invertible, in practice the estimates are extremely unstable for $p \approx n$, and inaccurate unless $p << n$. Thus having methods that allow us to overcome non-invertibility, instability and/or inaccuracy is desirable. For unsupervised methods, such as graphical models (Whittaker, 1990), widely used to analyze microarray data, undersampling also causes problems, since these methods also involve inversion of the sample variance-covariance matrix. Of course, there are methods that do not require inversion of the sample variance-covariance matrix, for example, classification and regression trees (Breiman et al, 1984) and support vector machines (Cristianini and Shawe-Taylor, 2000). Technically such methods can be applied when $p > n$, but they also suffer from undersampling: the produced results tend to be unreliable and highly unstable.

In this thesis we introduce an approach, which we call *augmented bootstrap*, that shares the spirit of many existing methods for improving estimation and inference by means of resampling and regularization techniques, so we start with a brief review of these methods. Historically, the first resampling techniques were jackknife and bootstrap. The jackknife was proposed in Quenouille (1956), and the bootstrap in Efron (1979). The jackknife allows us to estimate the bias and variance of an estimator, and the bootstrap provides an empirical approximation of its sampling distribution. Techniques derived from the bootstrap also lead to improved estimation efficiency. Essentially, the idea is to employ "additional information" generated by resampling to obtain more accurate estimates of the parameters of interest, or more generally better performing statistical procedures. An instance are *boosting* methods. The first boosting method proposed by Freund and Schapire

(1996) aggregates information from weighted samples drawn from the training data to improve classification performance or regression estimation. Reweighting of the data, i.e. a special resampling scheme, is dictated by performance at the current iteration, where performance is measured by some appropriate loss function. Another instance are *bagging* methods (shorthand for *bootstrap aggregating*). As introduced in Breiman (1996), the main idea of bagging is to compute an estimator on several (say B) bootstrap samples from the data, and produce an improved estimator by averaging the resulting estimates:

$$\theta^{bag}(X_1, \ldots, X_n) = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}(X_1^{(b)}, \ldots, X_n^{(b)}),$$

where $\hat{\theta}(\cdot)$ is any estimator, for example a "plug-in" estimator of the population statistical functional $\theta(F)$, and $X_1^{(b)}, \ldots, X_n^{(b)}$ are bootstrap samples drawn from the empirical distribution F_n of the original sample $X_1, \ldots, X_n$. For convenience we will denote the distribution corresponding to a bootstrap sample as $F_{n(h)}$, where n stands for the original sample size, and h is the size of the bootstrap sample. The most common version of bagging involves drawing B bootstrap samples of the same size as the original sample, with B typically fixed to be large enough, but computationally feasible. In fact, this commonly used version of bagging is a Monte-Carlo approximation to a more theoretical notion of bagging, where averaging occurs over all possible resamples of size n:

$$\begin{aligned} \theta^{bag}(F_{n(n)}) &\triangleq \theta^{bag}(X_1, \ldots, X_n) \\ &= \frac{1}{n^n} \sum_{b=1}^{n^n} \hat{\theta}(X_1^{(b)}, \ldots, X_n^{(b)}). \end{aligned}$$

The closer B is to n^n, the better the Monte-Carlo approximation, but for practical purposes using moderate B values proved to be effective (Breiman, 1996).

In full generality, the size of the bootstrap samples used in bagging does not have to equal the original sample size. Several papers considered subagging, where the size of the bootstrap samples is smaller than the original one (Knight and Bassett, 2002; Buhlmann, 2003). Subagging allows one to reduce computational time, and was shown to decrease the mean squared error with respect to unbagged

estimators in various applications. Utilizing bagging with a resampling size larger than the original one has also been proposed (Buja and Stuezle, 2006), and is of crucial importance for our augmented bootstrap method. Therefore, a bagged estimator can be formally defined as:

$$\begin{aligned} \theta^{bag}(F_{n(h)}) &= \frac{1}{n^h}\sum_{b=1}^{n^h}\hat{\theta}(X_1^{(b)},\ldots,X_h^{(b)}) \\ &= E[\hat{\theta}(X_1^{(b)},\ldots,X_h^{(b)})]. \end{aligned}$$

This definition was used in Buja and Stuezle (2006), whose derivations we follow in Chapter 2. Interestingly, bagging can be employed both for bootstrap samples with replacement, when $h \lessgtr n$, and for samples without replacement, for these $h < n$. For our purposes we consider bootstrap sampling with replacement.

Bagging was successfully used for constructing bagged classification and regression trees, and proved very useful in large dimensional complex problems. It was shown that the mean squared error of a bagged estimator can be smaller or equal to the mean squared error of the corresponding traditional plug-in estimator $\theta(F_n)$ (Breiman, 1996):

$$\begin{aligned} MSE(\theta^{bag}(F_{n(h)});\theta(F)) &= E[||\theta^{bag}(F_{n(h)}) - \theta(F)||^2] \\ &\leq E[||\theta(F_n) - \theta(F)||^2], \end{aligned}$$

where $||\cdot||$ indicates a norm for the space in which $\theta(\cdot)$ takes values. Details on the conditions under which bagging produces a mean square error improvement can be found, for example, in Friedman and Hall (2000), Buhlmann and Yu (2002), and Buja and Stuezle (2006).

One can think of bagging as a smoothing technique for improving unstable estimators, and this interpretation creates a link with the augmented bootstrap approach we are developing. The augmented bootstrap is based on resampling from a "noised" version of the data, so in addition to smoothing by averaging, it also smooths by composition with an appropriate perturbation.

This noising rationale creates another connection with a well-known method, namely the *smoothed bootstrap* (Efron, 1979; Silverman and Young, 1987). The

smoothed bootstrap advocates considering a "noised" (smoothed) version of the empirical distribution, for example:

$$F_n(\tau^2) = F_n \circ N(0, \tau^2\Sigma),$$

where $\circ$ indicates convolution, $\Sigma = Cov(F)$ is the variance-covariance matrix of the underlying population F, and τ^2 represents a smoothing parameter. Of course, in applications, instead of the unknown population Σ its sample version $\hat{\Sigma}$ is used. Silverman and Young (1987) studied the smoothed bootstrap for linear statistical functionals, i.e. functionals that can be expressed as:

$$\theta(F) = \int a(t)dF(t)$$

Theoretically, the smoothed bootstrap estimator can be defined in the following way:

$$\theta^{sm}(F_n(\tau^2)) \quad = \quad \int a(t)\hat{f}_\tau(t)dt,$$

where $\hat{f}_\tau(t)$ is a kernel estimate of the density function f that corresponds to the distribution F defined as:

$$\hat{f}_\tau(x) \quad = \quad |\Sigma|^{-\frac{1}{2}} \frac{1}{n} \sum_{i=1}^{n} K(\tau^{-1}\Sigma^{-\frac{1}{2}}(x - X_i)),$$

where $K(\cdot)$ is a so-called kernel function. A kernel function should be a symmetric density function with the same dimension as F and a unit variance-covariance matrix. The Gaussian density has the necessary properties, and is one of the popular kernel functions used in various applications. Simple algebraic calculations allow us to express the smoothed bootstrap estimator in the following way:

$$\theta^{sm}(F_n(\tau^2)) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} \omega(X_i),$$

where

$$\omega(x) \;=\; \int K(\xi)a(x+\tau\Sigma^{\frac{1}{2}}\xi)d\xi$$

In their paper, using Taylor series expansions Silverman and Young (1987) showed that $E[||\theta(F_n(\tau^2)) - \theta(F)||^2]$ can be made smaller than $E[||\theta(F_n) - \theta(F)||^2]$ for small values of τ^2, under conditions that, loosely speaking, require a negative association between $a(X)$ and its second derivatives. The same conditions guarantee improvement over $E[||\theta(F_n) - \theta(F)||^2]$ if the functional $\theta(\cdot)$ is locally linearizable in a von Mises sense (Silverman and Young, 1987; Fernholz, 1983) and the sample size is large, because in this case the behavior of $\theta^{sm}(F_n(\tau^2))$ is dominated by its linear approximation. Subsequent work by Hall et al (1989) and de Angelis and Young (1992) explored other conditions under which smoothing provides second order gains for estimation of quantile variances. In addition, Efron (1979) discussed improvements obtained via smoothing for small samples.

In practice, sampling from $F_n(\tau^2)$ is implemented resampling with replacement from F_n and adding to each point an independent draw from a Gaussian with mean 0 and variance-covariance matrix $\tau^2\hat{\Sigma}$. The work mentioned above implies that, in a variety of contexts, using noised resamples to produce an estimator $\theta^{sm}(F_n(\tau^2))$ of $\theta(F)$ can indeed lead to mean squared error improvements relative to the traditional plug-in estimator:

$$\begin{aligned} MSE(\theta^{sm}(F_n(\tau^2)); \theta(F)) &= E[||\theta^{sm}(F_n(\tau^2)) - \theta(F)||^2] \\ &\leq E[||\theta(F_n) - \theta(F)||^2]. \end{aligned}$$

1.2 A Motivating Application

A fascinating biological problem concerning the inactivation of genes on the X chromosome serves as the motivation for the line of research proposed in this thesis.

In mammalian species, females carry two copies of the X chromosome while males carry one copy of the X chromosome and one copy of the Y chromosome. Since only a small fraction of genes on chromosome X have Y-homologs, to ensure

that males and females have the same amount of X-linked gene products, a dosage-compensation mechanism comes into play. In each cell of a female, one copy of the X chromosomes is inactivated at random (Ross et al, 2005). Thus females are mosaics, meaning that in some of their cells the maternal X chromosome is expressed, and in others the paternal one is expressed. However, some of the genes on the inactivated X chromosome continue to be expressed to some degree, i.e. they escape inactivation. Many studies have investigated possible mechanisms and explanations for this phenomenon. One explanation relates inactivation status of genes to the distribution of L1 repeats in their genomic neighborhood (Lyon, 1998; Bailey et al, 2000), however this hypothesis remains controversial.

In a recent study (Carrel et al, 2006) we attempted to predict inactivation status using frequencies of overrepresented DNA motifs of certain lengths (k-mers) in windows around the transcription start sites (TSS) of the genes. In this study, we used k-mers of length 12 and windows ranging in size from 100 to 500 Kb (i.e. thousands of bases – positions). Because the DNA "alphabet" comprises four nucleotides (A, T, G, C), the number of possible k-mers of length 12 is very large; namely 4^{12}. We reduced this number dramatically by means of (1) a preliminary "variable selection" stage, based on testing for overrepresentation of k-mers in windows around inactivated (*subject*; Su) vs non-inactivated (*escape*; Es) genes, and (2) a "pooling" of similar k-mers into "meta"-k-mers (see Carrel et al (2006) for details). However, we were still left with $p = 248$ variables (frequencies of significantly overreppresented "meta"-k-mers). This is quite large compared to the number of genes whose inactivation status was experimentally assessed with enough confidence to use them as training data for our classifier.

We considered two classification problems. In the first one, we restricted attention to genes with validated status that are located in Xp22, a special region of chromosome X, plus a few genes outside Xp22 whose status was assessed by several independent studies. This gave us n between 42 and 93 genes, depending on the window size around the TSS (see Carrel et al (2006) for details). The choice of Xp22 was dictated by biological reasons, since this region is rich in genes that escape inactivation and has lower density of L1 repeats compared to the rest of the X chromosome. For the second classification, we used all X genes with validated status. This gave us n between 315 and 402 (again depending on the window

size). Thus, for the first classification we had $p > n$, and for the second $p < n$ but still relatively small – we had between 1.3 and 1.6 observations per predictor. To by-pass this problem, we applied principal components to the 248-dimensional frequency data, followed by a naïve bayes classifier on the SIR direction (Li, 1991) estimated restricting attention to the first five principal components.

Although this *ad hoc* strategy produced good classification performance (as measured by cross validation; see Table 1.1), the results of our study suggested that there would be ample room for improvement by using k-mer frequency data more effectively, and/or employing many other genomic features as predictors of the inactivation status of genes on the X chromosome.

Success rate	Xp22			whole-X		
	100 Kb	200 Kb	500 Kb	100 Kb	200 Kb	500 Kb
Es	85 %	90%	85%	72%	77%	84%
Su	93%	93%	93%	80%	91%	91%

Table 1.1. Leave-one-out cross validation success rates for the X inactivation problem. Es(cape) genes are genes that escape inactivation. Su(bject) genes are inactivated genes. 100, 200 and 500 Kb are sizes of windows around the genes' transcription start sites.

Using principal components prior to SIR led to loss of information potentially valuable for classification purposes. Using an iterative procedure akin to the one in Cook et al (2007), or the augmented bootstrap approach proposed here, will allow us to use all overrepresented "meta"-k-mers frequencies (not just a few of their principal components) in estimating directions relevant for predicting inactivation status, leading to better classification performance (see Section 3.3.1 for details).

1.3 The Augmented Bootstrap

In this section we introduce the general formulation of the augmented bootstrap methodology. We are interested in estimating a given population statistical functional defined on some distribution F; $\theta(F)$. At this stage we do not restrict the form for $\theta(F)$ in any fashion. For example, it could be a population variance-covariance matrix, or its inverse, or directions that span the central subspace in regression or classification problems, or a classification and/or regression tree, or

any other quantity or procedure of interest.

Suppose we observe an independently identically distributed (iid) sample $X_1, X_2, \ldots, X_n$ from the population F. The observations X_i will typically belong to $\mathbb{R}^p$. We denote the empirical distribution of such iid sample of size n as F_n. Let $\theta(F_n)$ be a plug-in estimator of $\theta(F)$. The accuracy of $\theta(F_n)$ will of course depend on n, and for some functionals (for example, for the inverse sample variance-covariance matrix) the relation of n to p may determine not just accuracy, but also whether the plug-in estimator can be computed in the first place. The problem, of course, is that we cannot draw more observations from F.

However, suppose we could draw $h = mn$ independent observations from a noised version of our population, where h can be made arbitrarily large (in particular, larger than n). We indicate this noised distribution as $F(\tau^2)$, where τ^2 is a parameter (scalar or vector) indexing the amount of noise. In our applications we will typically use convolution with a spherical Gaussian, say $N(0, \tau^2 I)$. Unfortunately, we can not sample from $F(\tau^2)$ directly. However, we can approximate this by sampling from the noised version of the empirical distribution of the observed sample, which we indicate with $F_n(\tau^2)$. We denote the empirical distribution of the resulting sample as $F_{n(mn)}(\tau^2)$. In practice we will take a bootstrap sample (with replacement) of size mn from F_n, and add iid draws from the "noising" distribution $N(0, \tau^2 I)$. We call an estimator computed on such sample an *augmented bootstrap estimator*, and denote it as $\theta^{AB}(F_{n(mn)}(\tau^2))$. As we will see in the following Chapters, "noising" allows us, if needed, to generate a number of *distinct* (pseudo) observations in excess of p also when $p > n$, which is useful in many applications. The augmented bootstrap provides a way to handle non-invertibility or instability in the inversion of a sample variance-covariance matrix when estimating Σ^{-1}. This is useful for both unsupervised and supervised problems; among the latter, for estimating directions in the central subspace, as required by methods for sufficient dimension reduction (Li, 1991; Cook, 1998; Chiaromonte et al, 2002). The augmented bootstrap also helps to ameliorate overfitting for classification and regression trees (Breiman et al, 1984, CART). Details on the augmented bootstrap for these applications are considered in Chapters 3 and 4.

1.4 Overview of the following chapters

The rest of this thesis is organized in the following manner. In Chapter 2 we present some theoretical developments for the augmented bootstrap method. Similarly to developments in Buja and Stuezle (2006), we consider statistical functionals that comprise sums of V-statistics. We are mostly interested in super-sampling, i.e. the case when bootstrap samples drawn to obtain a bagged estimator have size $h = mn$, $m > 1$ larger than the original sample size n. Reproducing results from Buja and Stuezle (2006), we study the mean squared error of the bagged estimator as a function of the reciprocal of the sample size multiplier ($g = 1/m$) and restate the conditions under which bagging allows to reduce the mean squared error with respect to the plug-in estimator. Next we extend the results from Silverman and Young (1987) to a sum of V-statistics, finding conditions under which using the smoothed bootstrap will reduce the mean squared error for small values of the smoothing standard deviation τ. Then we define our augmented bootstrap procedure as a composition of bagging and smoothing. We study the shape and behavior of the mean squared error of the augmented bootstrap estimator as function of two tuning parameters: the reciprocal of the sample size multiplier (g) and the smoothing standard deviation (τ). Results concerning the behavior of the mean squared error for bagged and smoothed estimators help us establish conditions under which the augmented bootstrap estimator can achieve a reduction in mean squared error relative to the plug-in, smoothed and/or bagged estimators.

In Chapter 3 we consider the important statistical problem of estimating an inverse variance-covariance matrix. We assess the performance of the augmented bootstrap method in comparisons to the other widely applied methods such as Moore-Penrose inverse, shrinkage prior to inversion, bagging and so on. We consider a variety of simulation scenarios and several applications to genomic data. We revisit our motivating example on X-chromosome inactivation from Section 1.2, and look at another interesting biological application that concerns analyzing data from microarray experiments. Moreover, in this chapter we discuss the tight connections of the augmented bootstrap method with ridge regularization, and explore possible modifications of the augmented bootstrap for estimating inverse variance-covariance matrices.

While inversion of the sample variance-covariance matrix requires special attention when sample sizes are small (the sample variance-covariance matrix can not be inverted when the number of observations is smaller than the dimension, and the inversion is unstable and/or unreliable when the number of observations exceeds the dimension but is still small) many statistical methods do not explicitly require large sample sizes. For example, classification and regression trees, which we consider in Chapter 4 can, in principle, be computed when the dimension of the data is larger than the sample size. However, also here, data scarcity causes unstable and unreliable results. One of the major problems for such applications is indeed overfitting. We study the behavior of the augmented bootstrap method as an aid to mitigate overfitting, and compare its performance to methods such as recursive shrinkage and bagging.

In Chapter 5 we summarize our developments, provide some concluding remarks and give a brief overview of future research directions.

CHAPTER 2

Some analytical results concerning the augmented bootstrap

2.1 Preliminaries

In this chapter we will follow the lines of reasoning from the literature on bagging and smoothed bootstrap (Buja and Stuezle, 2006; Silverman and Young, 1987) and for simplicity of calculations restrict our attention to one dimensional statistical functionals of the distribution F having the form:

$$\begin{aligned} \theta(F) &= \int A(x_1)dF(x_1) + \int B(x_1,x_2)dF(x_1,x_2) \\ &+ \int C(x_1,x_2,x_3)dF(x_1,x_2,x_3) + \dots \end{aligned} \tag{2.1}$$

It is easy to see that the plug-in estimator for functional 2.1 calculated from the iid sample $X_1, X_2, \dots, X_n$, with empirical distribution F_n, will be:

$$\begin{aligned}\theta(F_n) &= \frac{1}{n}\sum_i A(X_i) + \frac{1}{n^2}\sum_{i,j} B(X_i, X_j) \\ &+ \frac{1}{n^3}\sum_{i,j,k} C(X_i, X_j, X_k) + \dots \end{aligned} \tag{2.2}$$

Without loss of generality we can assume that the terms $B(\cdot,\cdot)$, $C(\cdot,\cdot,\cdot)$, ... are permutation symmetric. Permutation symmetry allows us to keep derivations more compact notation-wise. If the terms are not permutation symmetric, to gain permutation symmetry we can always transform them by averaging over all permutations of the arguments. This transformation will not change the properties of the functionals $\theta(F)$ and $\theta(F_n)$.

In the following subsections we briefly summarize results from Buja and Stuezle (2006) on bagging statistical functionals of the form 2.1, and adapt results on the smoothed bootstrap from Silverman and Young (1987) to functionals of such form. For simplicity, we work with functionals of the second order, i.e. functionals that only include terms involving $A(\cdot)$ and $B(\cdot,\cdot)$.

2.1.1 Bagging

Suppose we have an iid sample $X_1, X_2, \dots, X_n$ from a distribution F with empirical distribution F_n. We consider a bootstrap sample (with replacement) of size $h \lessgtr n$ from the original sample, $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_h$, and denote its empirical distribution as $\tilde{F}_h$. Then the plug-in estimator of 2.1 calculated on the resample can be written as:

$$\begin{aligned}\theta(\tilde{F}_h) &= \frac{1}{h}\sum_{i=1}^{h} A(\tilde{X}_i) + \frac{1}{h^2}\sum_{i,j=1}^{h} B(\tilde{X}_i, \tilde{X}_j) \\ &= \frac{1}{h}\sum_{i=1}^{n} W_i A(X_i) + \frac{1}{h^2}\sum_{i,j=1}^{n} W_i W_j B(X_i, X_j), \end{aligned} \tag{2.3}$$

where W_i, $i = 1, \dots, n$, are the multiplicities with which the X_i's from the original sample appear in the resample. As reviewed in Chapter 1, bagging here is meant as averaging over all bootstrap samples of size h from $X_1, X_2, \dots, X_n$. We will denote

the bagged estimator as $\theta^{bag}(F_{n(h)})$. For equation 2.3 bagging corresponds to taking an expectation over the multiplicities W_i. As argued in Buja and Stuezle (2006), when sampling with replacement these multiplicities have multinomial distribution with h trials and probability vector $(1/n, 1/n, \ldots, 1/n)$. Therefore:

$$\begin{aligned}
\theta^{bag}(F_{n(h)}) &= E(\theta(\tilde{F}_h)) \\
&= E\left(\frac{1}{h}\sum_{i=1}^{n} W_i A(X_i) + \frac{1}{h^2}\sum_{i,j=1}^{n} W_i W_j B(X_i, X_j)\right) \\
&= \frac{1}{h}\sum_{i=1}^{n} E(W_i) A(X_i) + \frac{1}{h^2}\sum_{i,j=1}^{n} E(W_i W_j) B(X_i, X_j) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(A(X_i) + \frac{1}{h}B(X_i, X_i)\right) + \frac{1}{n^2}\sum_{i,j=1}^{n}\left(1 - \frac{1}{h}\right) B(X_i, X_j),
\end{aligned}$$

because $E(W_i) = \frac{h}{n}$, $E(W_i, W_j) = \frac{h(h-1)}{n^2}$ when $i \neq j$, and $E(W_i^2) = \frac{h}{n} + \frac{h(h-1)}{n^2}$. Suppose that the resample size is $h = mn$; we call m a *sample size multiplier*. Then we can rewrite the bagged estimator as a function of the reciprocal of the sample size multiplier, $g = 1/m$, as follows:

$$\theta^{bag}(F_{n(mn)}) = \theta(F_n) + g\mathcal{A}_{bag}, \tag{2.4}$$

where

$$\mathcal{A}_{bag} = \frac{1}{n^2}\sum_{i=1}^{n} B(X_i, X_i) - \frac{1}{n^3}\sum_{i,j=1}^{n} B(X_i, X_j),$$

and $\theta(F_n)$ is the plug-in estimator as defined in equation 2.2. Next we study the behavior of the mean squared error of the bagged estimator as a function of g, reproducing the developments in Buja and Stuezle (2006). We restrict our attention only to the terms that contain n^{-1} and n^{-2}, and collect all higher order terms into $\mathcal{O}(n^{-3})$. Details concerning the calculations and the notation used in the following proposition are given in Appendix A.1.

Proposition 2.1.1. *(adapted from Buja & Stuezle)*
The mean squared error of $\theta^{bag}(F_{n(mn)})$ *can be written as:*

$$\begin{aligned} MSE(\theta^{bag}(F_{n(mn)}); g) &= MSE(\theta(F_n)) + g(\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)}) \\ &+ g^2(\beta_{bias}^{(bag)}) + \mathcal{O}(n^{-3}), \end{aligned}$$

where

$$\begin{aligned} \alpha_{var}^{(bag)} &= \frac{1}{n^2} Cov(A_X + 2B_X; B_{XX} - 2B_X) \\ \beta_{bias}^{(bag)} &= \frac{1}{n^2} \left(E(B_{XX}) - E(B_{XY})\right)^2. \end{aligned}$$

Proof. See Appendix A.1 for relevant calculations and definitions of the terms A_X, B_X, B_{XX}, etc. □

Note that here and throughout the superscripts denote whether the coefficients correspond to variance or bias squared terms in the MSE decomposition.

Thus, the mean squared error of the bagged estimator, when viewed as a function of g, constitutes a parabola. At $g = 0$, which corresponds to an infinite sample size multiplier, $MSE(\theta^{bag}(F_{n(mn)}); 0) = MSE(\theta(F_n))$. For a certain range of g we can achieve a reduction in the mean squared error of the bagged estimator relative to the plug-in estimator, when $MSE(\theta^{bag}(F_{n(mn)}); g)$ has a minimum on $\mathbb{R}^+$.

Proposition 2.1.2. *(adapted from Buja & Stuezle)*
If $\beta_{bias}^{(bag)} > 0$ *and* $\alpha_{bias}^{(bag)} + \alpha_{var}^{(bag)} < 0$, *then*

$$MSE(\theta^{bag}(F_{n(mn)}); g) < MSE(\theta(F_n)) \quad , \quad \forall g \in \left(0, -\frac{\alpha_{bias}^{(bag)} + \alpha_{var}^{(bag)}}{\beta_{bias}^{(bag)}}\right). \quad (2.5)$$

Proof. We have:

$$\frac{\partial MSE(\theta^{bag}(F_{n(mn)}))}{\partial g} = \alpha_{bias}^{(bag)} + \alpha_{var}^{(bag)} + 2g\beta_{bias}^{(bag)}$$

Thus, the unique root of the first derivative is equal to $g^* = -\frac{\alpha_{bias}^{(b)} + \alpha_{var}^{(b)}}{2\beta_{bias}^{(b)}}$. Under the conditions on the coefficients, $g^* > 0$, $\frac{\partial^2 MSE(\theta^{bag}(F_{n(mn)}))}{\partial g^2}|_{g=g^*} = 2\beta_{bias}^{(b)} > 0$, and therefore g^* is a minimum. □

While the condition $\beta_{bias}^{(bag)} > 0$ is always satisfied for functionals of the form 2.2, the condition $\alpha_{bias}^{(bag)} + \alpha_{var}^{(bag)} < 0$ poses constraints on the bias-variance tradeoff. The bagged estimator has smaller mean squared error than the plug-in estimator when the increase in bias induced by bagging is smaller than the corresponding decrease in variance. In turn, the decrease in variance occurs whenever $Cov(A_X + 2B_X; B_{XX} - 2B_X) < 0$. Note that $g \in (0,1)$ corresponds to $m \in (1,\infty)$, i.e. to a bootstrap sample size m greater than the original sample size n. Thus, when the minimum of the MSE is achieved between 0 and 1, bagged estimators with resampling sizes greater than n will beat the plug-in estimator in terms of mean squared error.

2.1.2 Smoothing

Here we extend results of Silverman and Young (1987) on the smoothed bootstrap to functionals of the form 2.1. We formally define a smoothed bootstrap estimator based on an iid sample $X_1, X_2, \ldots, X_n$ as:

$$\theta^{sm}(F_n(\tau^2)) = \frac{1}{n}\sum_{i=1}^{n}\omega_1(X_i) + \frac{1}{n^2}\sum_{i,j=1}^{n}\omega_2(X_i, X_j), \tag{2.6}$$

where:

$$\begin{aligned}\omega_1(X_i) &= \int A(X_i + \tau\xi_1)\phi(\xi_1)d\xi_1 \\ \omega_2(X_i, X_j) &= \int B(X_i + \tau\xi_1, X_j + \tau\xi_2)\phi(\xi_1, \xi_2)d\xi_1 d\xi_2\end{aligned}$$

and the integration is against Gaussian densities; more precisely, $\phi(\xi)$ is a standard normal density, and $\phi(\xi_1, \xi_2)$ a bivariate normal density with mean $(0,0)$ and identity variance-covariance matrix I. We call τ^2 a *smoothing variance*, and τ a *smoothing standard deviation*.

To study the behavior of the mean squared error of the smoothed bootstrap estimator as a function of τ, we employ Taylor expansions of the functionals $A(x_i + \tau\xi_1)$ and $B(x_i + \tau\xi_1, x_j + \tau\xi_2)$ around x_i and (x_i, x_j), respectively. The Taylor

expansions give:

$$\begin{aligned}
A(x_i+\tau\xi_1) &= A(x_i)+\tau\xi\frac{\partial A}{\partial x_i}(x_i)+\frac{1}{2}\tau^2\xi^2\frac{\partial^2 A}{\partial x_i^2}(x_i)\\
&+ \frac{1}{6}\tau^3\xi^3\frac{\partial^3 A}{\partial x_i^3}(x_i)+\frac{1}{24}\tau^4\xi^4\frac{\partial^4 A}{\partial x_i^4}(x_i)+\mathcal{O}(\tau^5)\\
B(x_i+\tau\xi_1,x_j+\tau\xi_2) &= B(x_i,x_j)+\left[\tau\xi_1\frac{\partial B}{\partial x_i}(x_i,x_j)+\tau\xi_2\frac{\partial B}{\partial x_j}(x_i,x_j)\right]\\
&+ \frac{1}{2}[\tau^2\xi_1^2\frac{\partial^2 B}{\partial x_i^2}(x_i,x_j)+2\tau^2\xi_1\xi_2\frac{\partial^2 B}{\partial x_i\partial x_j}(x_i,x_j)\\
&+ \tau^2\xi_2^2\frac{\partial^2 B}{\partial x_j^2}(x_i,x_j)]\\
&+ \frac{1}{6}[\tau^3\xi_1^3\frac{\partial^3 B}{\partial x_i^3}(x_i,x_j)+3\tau^3\xi_1^2\xi_2\frac{\partial^3 B}{\partial x_i^2\partial x_j}(x_i,x_j)\\
&+ 3\tau^3\xi_1\xi_2^2\frac{\partial^3 B}{\partial x_i\partial x_j^2}(x_i,x_j)+\tau^3\xi_2^3\frac{\partial^3 B}{\partial x_j^3}(x_i,x_j)]\\
&+ \frac{1}{24}[\tau^4\xi_1^4\frac{\partial^4 B}{\partial x_i^4}(x_i,x_j)+4\tau^4\xi_1^3\xi_2\frac{\partial^4 B}{\partial x_i^3\partial x_j}(x_i,x_j)\\
&+ 6\tau^4\xi_1^2\xi_2^2\frac{\partial^4 B}{\partial x_i^2\partial x_j^2}(x_i,x_j)+4\tau^4\xi_1\xi_2^3\frac{\partial^4 B}{\partial x_i\partial x_j^3}(x_i,x_j)\\
&+ \tau^4\xi_2^4\frac{\partial^4 B}{\partial x_j^4}(x_i,x_j)]+\mathcal{O}(\tau^5)
\end{aligned}$$

Therefore, taking into account the permutation symmetry of $B(\cdot,\cdot)$, integration against Gaussian densities produces:

$$\begin{aligned}
\omega_1(x_i) &= A(x_i)+\frac{1}{2}\tau^2\frac{\partial^2 A}{\partial x_i^2}(x_i)+\frac{1}{8}\tau^4\frac{\partial^4 A}{\partial x_i^4}(x_i)+\mathcal{O}(\tau^5)\\
\omega_2(x_i,x_j) &= B(x_i,x_j)+\tau^2\frac{\partial^2 B}{\partial x_i^2}(x_i,x_j)+\frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^4}(x_i,x_j)\\
&+ \frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^2\partial x_j^2}(x_i,x_j)+\mathcal{O}(\tau^5)
\end{aligned}$$

Thus, the smoothed bootstrap estimator can be approximated as:

$$\theta^{sm}(F_n(\tau^2)) = \theta(F_n)+\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\tau^2\frac{\partial^2 A}{\partial x_i^2}(x_i)+\frac{1}{8}\tau^4\frac{\partial^4 A}{\partial x_i^4}(x_i)\right)$$

$$
\begin{aligned}
&+ \quad \frac{1}{n^2}\sum_{i,j=1}^{n}\left(\tau^2\frac{\partial^2 B}{\partial x_i^2}(x_i,x_j)+\frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^4}(x_i,x_j)+\frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^2\partial x_j^2}(x_i,x_j)\right)\\
&+ \quad \mathcal{O}(\tau^5) \qquad (2.7)
\end{aligned}
$$

We can write $\theta^{sm}(F_n(\tau^2))$ as a function of τ:

$$\theta^{sm}(F_n(\tau^2)) = \theta(F_n) + \tau^2\mathcal{A}_{sm} + \tau^4\mathcal{B}_{sm} + \mathcal{O}(\tau^5), \qquad (2.8)$$

where:

$$
\begin{aligned}
\mathcal{A}_{sm} &= \frac{1}{2}\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 A}{\partial X_i^2}(X_i) + \frac{1}{n^2}\sum_{i,j=1}^{n}\frac{\partial^2 B}{\partial X_i^2}(X_i,X_j)\\
\mathcal{B}_{sm} &= \frac{1}{8}\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^4 A}{\partial X_i^4}(X_i) + \frac{1}{4}\frac{1}{n^2}\sum_{i,j=1}^{n}\left(\frac{\partial^4 B}{\partial X_i^4}(X_i,X_j)+\frac{\partial^4 B}{\partial X_i^2 X_j^2}(X_i,X_j)\right)
\end{aligned}
$$

The derivation for $MSE(\theta^{sm}(F_n(\tau^2));\tau)$ is similar to that for the bagged estimator in Section 2.1.1. Consequently, we have the following proposition:

Proposition 2.1.3. *The mean squared error of the smoothed bootstrap estimator $\theta^{sm}(F_n(\tau^2))$ can be written as:*

$$
\begin{aligned}
MSE(\theta^{sm}(F_n(\tau^2));\tau) &= MSE(\theta(F_n)) + \tau^2(\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)})\\
&+ \tau^4(\beta_{bias}^{(sm)} + \beta_{var}^{(sm)}) + \mathcal{O}(n^{-3}) + \mathcal{O}(\tau^5), \qquad (2.9)
\end{aligned}
$$

where $\alpha_{bias}^{(sm)}$, $\alpha_{var}^{(sm)}$, $\beta_{bias}^{(sm)}$, and $\beta_{var}^{(sm)}$ are defined in Appendix A.2.

The coefficients in the above proposition contain sums of the covariances between $A(\cdot)$, $B(\cdot,\cdot)$ and their derivatives up to the fourth order.

The plausible range of values for the smoothing standard deviation is $\mathbb{R}^+$. The mean squared error of the smoothed bootstrap estimator is a polynomial function in τ, and when no smoothing occurs ($\tau = 0$) it is equal to the mean squared error of the plug-in estimator: $MSE(\theta^{sm}(F_n(\tau^2));0) = MSE(\theta(F_n))$. To get a reduction (with respect to the plug-in estimator) in the mean squared error due to smoothing, $MSE(\theta^{sm}(F_n(\tau^2));\tau)$ should be a decreasing function, at least for small positive values of τ.

Proposition 2.1.4. *If* $\beta_{bias}^{(sm)} + \beta_{var}^{(sm)} > 0$ *and* $\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)} < 0$, *then*

$$MSE(\theta^{sm}(F_n(\tau^2)); \tau) < MSE(\theta(F_n)). \tag{2.10}$$

for small enough values of τ.

Proof. We have:

$$\frac{\partial MSE(\theta^{sm}(F_n(\tau^2)))}{\partial \tau} = 2\tau(\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)}) + 4\tau^3(\beta_{bias}^{(sm)} + \beta_{var}^{(sm)})$$

The positive root of the first derivative is equal to $\tau^* = \sqrt{-\frac{\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)}}{2(\beta_{bias}^{(sm)} + \beta_{var}^{(sm)})}}$. Thus, under the conditions on the coefficients, $\tau^* > 0$,

$$\frac{\partial^2 MSE(\theta^{sm}(F_n(\tau^2)); \tau)}{\partial \tau^2}|_{\tau=\tau^*} = -10(\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)}) > 0$$

and therefore τ^* is a minimum. It follows that $MSE(\theta^{sm}; \tau)$ is a decreasing function on $(0, \tau^*)$. □

It is important to remark that, in practice, the optimal value of τ could be different from the τ^* in proposition 2.1.4 because of the term $\mathcal{O}(\tau^5)$ in equation 2.9. Also, for large values of τ our Taylor approximations might be invalid.

In their paper on the smoothed bootstrap Silverman and Young (1987) consider only linear functionals (i.e. functionals of the form 2.1 that involve only terms with $A(\cdot)$). They find conditions under which smoothing provides a reduction in the mean squared error relative to the plug-in estimator. If the covariance between $A(X)$ and its second derivative is negative (i.e. $Cov(A_X, A_X^{(2)}) < 0$), then smoothing is beneficial (see Appendix A.2 for the definitions of A_X, $A_X^{(2)}$, and additional details). The main difference with our derivation is that they employ a Taylor expansion up to the second order only, and therefore the expression for the mean squared error contains only terms up to τ^2, with higher order terms collected into $\mathcal{O}(\tau^4)$. In fact, when we disregard terms containing higher powers of τ and terms that contain $B(\cdot, \cdot)$ and its derivatives, our conditions on the coefficients indeed reduce to $Cov(A_X, A_X^{(2)}) < 0$. This covariance is contained as a summand in the expression for the variance of the smoothed bootstrap estimator; more precisely, it is contained in $\alpha_{var}^{(sm)}$.

2.2 The Augmented Bootstrap

In this section, paralleling the developments for bagging and smoothing, we define the theoretical formulation of the augmented bootstrap estimator $\theta^{AB}(F_{n(mn)}(\tau^2))$ and study properties of its mean squared error as a function of the smoothing standard deviation τ and the reciprocal of sample size multiplier g. We think of the augmented bootstrap in the following way: suppose we have an iid sample $X_1, X_2, \ldots, X_n$ from a distribution F, take a bootstrap resample of size $h = mn \geq n$ from it, $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_h$, and add independent draws from a Gaussian noise with mean 0 and variance-covariance $\tau^2 I$ to each resample element. This way we treat our augmented bootstrap estimator as a smoothed bootstrap estimator based on a "multiplied" (i.e. augmented) resample. We can approximate it with Taylor expansions as in 2.7, rewriting its expression in terms of the original sample $X_1, X_2, \ldots, X_n$, and take expectation over multinomially distributed multiplicities similarly to 2.4. We therefore obtain:

$$\begin{aligned}
\theta^{AB}(F_{n(mn)}(\tau^2)) &= \theta(F_n) + g\left[\frac{1}{n^2}\sum_{i=1}^{n} B(X_i, X_i) - \frac{1}{n^3}\sum_{i,j=1}^{n} B(X_i, X_j)\right] \\
&+ \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\tau^2\frac{\partial^2 A}{\partial x_i^2}(X_i) + \frac{1}{8}\tau^4\frac{\partial^4 A}{\partial x_i^4}(X_i)\right) \\
&+ \frac{1}{n^2}\sum_{i,j=1}^{n}(\tau^2\frac{\partial^2 B}{\partial x_i^2}(X_i, X_j) + \frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^4}(X_i, X_j) \\
&+ \frac{1}{4}\tau^4\frac{\partial^4 B}{\partial x_i^2 \partial x_j^2}(X_i, X_j)) \\
&+ g\tau^2\left(\frac{1}{n^2}\sum_{i=1}^{n}\frac{\partial^2 B}{\partial x_i^2}(X_i, X_i) - \frac{1}{n^3}\sum_{i,j=1}^{n}\frac{\partial^2 B}{\partial x_i^2}(X_i, X_j)\right) \\
&+ g\tau^4\frac{1}{4}(\frac{1}{n^2}\sum_{i=1}^{n}\left[\frac{\partial^4 B}{\partial x_i^4}(X_i, X_i) + \frac{\partial^4 B}{\partial x_i^2 \partial x_j^2}(X_i, X_i)\right] \\
&- \frac{1}{n^3}\left[\frac{\partial^4 B}{\partial x_i^4}(X_i, X_j) + \frac{\partial^4 B}{\partial x_i^2 \partial x_j^2}(X_i, X_j)\right]) + \mathcal{O}(\tau^5) \qquad (2.11)
\end{aligned}$$

Comparing the augmented bootstrap estimator to expressions 2.4 and 2.7 we observe that it can be decomposed into four parts: the plug-in estimator, a part arising from bagging, a part arising from smoothing, and a part that appears due

to the interplay between bagging and smoothing:

$$\begin{aligned}\theta^{AB}(F_{n(mn)}(\tau^2)) &= \theta(F_n) + g\mathcal{A}_{bag} + \tau^2\mathcal{A}_{sm} + \tau^4\mathcal{B}_{sm} \\ &+ g\tau^2\mathcal{A}_{inter} + g\tau^4\mathcal{B}_{inter} + \mathcal{O}(\tau^5), \end{aligned} \tag{2.12}$$

where

$$\begin{aligned}\mathcal{A}_{inter} &= \frac{1}{n^2}\sum_{i=1}^{n}\frac{\partial^2 B}{\partial x_i^2}(X_i, X_i) - \frac{1}{n^3}\sum_{i,j=1}^{n}\frac{\partial^2 B}{\partial x_i^2}(X_i, X_j) \\ \mathcal{B}_{inter} &= \frac{1}{4}(\frac{1}{n^2}\sum_{i=1}^{n}\left[\frac{\partial^4 B}{\partial x_i^4}(X_i, X_i) + \frac{\partial^4 B}{\partial x_i^2 \partial x_j^2}(X_i, X_i)\right] \\ &- \frac{1}{n^3}\left[\frac{\partial^4 B}{\partial x_i^4}(X_i, X_j) + \frac{\partial^4 B}{\partial x_i^2 \partial x_j^2}(X_i, X_j)\right])\end{aligned}$$

and $\mathcal{A}_{bag}$, $\mathcal{A}_{sm}$, $\mathcal{B}_{sm}$ are as defined in subsections 2.1.1 and 2.1.2. Correspondingly, we obtain the following decomposition for the mean squared error:

Proposition 2.2.1. *The mean squared error of* $\theta^{AB}(F_{n(mn)}(\tau^2))$ *can be written as:*

$$\begin{aligned}MSE(\theta^{AB}(F_{n(mn)}(\tau^2)); g, \tau) &= MSE(\theta(F_n)) + g(\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)}) + g^2(\beta_{bias}^{(bag)}) \\ &+ \tau^2(\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)}) + \tau^4(\beta_{bias}^{(sm)} + \beta_{var}^{(sm)}) + \\ &+ g\tau^2(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + g\tau^4(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)}) \\ &+ g^2\tau^2\delta_{bias}^{(inter)} + g^2\tau^4\gamma_{bias}^{(inter)} \\ &+ \mathcal{O}(n^{-3}) + \mathcal{O}(\tau^5), \end{aligned} \tag{2.13}$$

where the coefficients $\alpha_{bias}^{(inter)}$, $\alpha_{var}^{(inter)}$, $\beta_{bias}^{(inter)}$, $\beta_{var}^{(inter)}$, $\delta_{bias}^{(inter)}$ *and* $\gamma_{bias}^{(inter)}$ *are defined in Appendix A.3.*

Like the augmented bootstrap estimator itself (Equation 2.12), its MSE has four parts: the MSE of the plug-in estimator, terms from the MSE of the bagged estimator, terms from the MSE of the smoothed bootstrap estimator, and terms that appear due to the interplay between bagging and smoothing.

Next we study the mean squared error of the augmented bootstrap estimator as a function of its two tuning parameters; namely the smoothing standard deviation τ and the reciprocal of sample size multiplier $g = 1/m$.

Proposition 2.2.2. *The mean squared error of $\theta^{AB}(F_{n(mn)}(\tau^2))$ is a function of both g and τ:*

1. *For any fixed τ, if $\beta_{bias}^{(bag)} + \tau^2\delta_{bias}^{(inter)} + \tau^4\gamma_{bias}^{inter} > 0$ and $\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)} + \tau^2(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + \tau^4(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)}) < 0$, then*

$$MSE(\theta^{AB}(F_{n(mn)}(\tau^2)); g) < MSE(\theta^{sm}(F_n(\tau^2)))$$

   for all $g \in \left(0; -2\frac{\beta_{bias}^{(bag)}+\tau^2\delta_{bias}^{(inter)}+\tau^4\gamma_{bias}^{(inter)}}{\beta_{bias}^{(bag)}+\alpha_{var}^{(bag)}+\tau^2(\alpha_{bias}^{(inter)}+\alpha_{var}^{(inter)})+\tau^4(\beta_{bias}^{(inter)}+\beta_{var}^{(inter)})}\right)$

2. *For any fixed g, if $\beta_{bias}^{(sm)} + \beta_{var}^{(sm)} + g(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)}) + g^2\gamma_{bias}^{(inter)} > 0$ and $\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)} + g(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + g^2\delta_{bias}^{(inter)} < 0$, then*

$$MSE(\theta^{AB}(F_{n(mn)}(\tau^2)); \tau) < MSE(\theta^{bag}(F_{n(mn)}))$$

   for small enough values of τ.

Proof. 1. When viewed as a function of g, $MSE(\theta^{(AB)}(F_n); g, \tau)$ can be written as:

$$\begin{aligned} & MSE(\theta^{sm}(F_n(\tau^2))) \\ + \;& g\left(\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)} + \tau^2(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + \tau^4(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)})\right) \\ + \;& g^2\left(\beta_{bias}^{(bag)} + \tau^2\delta_{bias}^{(inter)} + \tau^4\gamma_{bias}^{(inter)er}\right) \\ + \;& \mathcal{O}(n^{-3}) + \mathcal{O}(\tau^5), \end{aligned} \tag{2.14}$$

   Conditions on coefficients are derived as in proposition 2.1.2.

2. When viewed as a function of τ, $MSE(\theta^{(AB)}(F_n); g, \tau)$ can be written as:

$$\begin{aligned} & MSE(\theta^{bag}(F_{n(mn)})) \\ + \;& \tau^2(\alpha_{bias}^{(sm)} + \alpha_{var}^{(sm)} + g(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + g^2\delta_{bias}^{(inter)}) \\ + \;& \tau^4(\beta_{bias}^{(sm)} + \beta_{var}^{(sm)} + g(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)}) + g^2\gamma_{bias}^{(inter)}) \\ + \;& \mathcal{O}(n^{-3}) + \mathcal{O}(\tau^5), \end{aligned} \tag{2.15}$$

Conditions on the coefficients are derived as in proposition 2.1.4.

□

Interestingly, the shape of the MSE of our augmented bootstrap estimator, when viewed as a function g, coincides with the shape of $MSE(\theta^{bag}(F_{n(mn)}); g)$, and when $g = 0$ (i.e. when the sample size multiplier is infinite):

$$MSE(\theta^{AB}(F_{n(mn)}(\tau^2)); 0) = MSE(\theta^{sm}(F_n(\tau^2))).$$

Conversely, when we consider the MSE as a function of τ, its shape coincides with the shape of $MSE(\theta^{sm}(F_n(\tau^2)))$; when $\tau = 0$ (i.e. when no smoothing occurs)

$$MSE(\theta^{AB}(F_{n(mn)}(\tau^2)); 0) = MSE(\theta^{bag}(F_{n(mn)}).$$

However, the performance of the augmented bootstrap depends also on the interaction terms, which appear due to the interplay between bagging and smoothing. Consequently, an overall reduction in the MSE relative to the plug-in estimator might be observed even when both bagging and smoothing have, separately, deleterious effects, since advantages from their interaction can compensate for them. Of course when bagging and/or smoothing improve estimation, their composition, as prescribed in the augmented bootstrap, can further decrease the MSE.

2.3 Concluding remarks

In this chapter we presented derivations for functionals of the form 2.1 involving only first, $A(\cdot)$, and second, $B(\cdot, \cdot)$, degree terms. It is quite straightforward to generalize these derivations to functionals that also involve terms with higher degree (i.e. $C(\cdot, \cdot, \cdot)$, $D(\cdot, \cdot, \cdot, \cdot)$ and so on), as described in Buja and Stuezle (2006). Both for bagging and smoothing, and therefore for the augmented bootstrap, the coefficients appearing in the mean squared error will, correspondingly, involve sums of covariances between these higher degree terms and their derivatives. Moreover, as noted in Buja and Stuezle (2006), functionals with finite von Mises expansions (or functionals well approximated by a finite von Mises expansion) can be written in the form 2.1. Therefore, the results we presented for bagging, smoothing and

the augmented bootstrap will generalize to these functionals.

Finally, generalization to multivariate functionals should be possible as well. Bagging for multivariate functionals does not require any additional consideration, and employing multivariate Taylor expansions will allow us to describe the effects of smoothing. It must be noted, though, that the mean squared error of multidimensional estimators needs to be defined in terms of an appropriate distance or norm in the relevant space (see Chapter 3).

CHAPTER 3

Augmented bootstrap for inverse variance-covariance matrices

3.1 Introduction

The material in this Chapter follows closely our developments as published in Tyekucheva and Chiaromonte (2008a) and Tyekucheva and Chiaromonte (2008b).

As mentioned in Chapter 1, many questions of biological interest can be addressed by supervised analyses in which a response Y (quantitative or categorical) is studied as a function of $X \in \mathbb{R}^p$, whose dimension one wishes to reduce prior to the fitting of regression equations or classifiers. Other questions of biological interest concern the interdependence structure among the coordinates of $X \in \mathbb{R}^p$, and can be addressed using various types of network models. Both supervised dimension reduction and network modeling, as well as many other statistical techniques ubiquitously employed in the analysis of genomic data, require estimation of the inverse variance-covariance matrix of the features under consideration, say $\Sigma = Cov(F)$, where F indicates the distribution of X on $\mathbb{R}^p$. Methods are therefore

needed to handle the estimation of Σ^{-1} when the number of features p is large relative to the number of available observations n. In these settings, inversion of the sample variance-covariance matrix $Cov(F_n)$, where F_n indicates the empirical distribution, can be impossible – when $n < p$, $Cov(F_n)$ is singular. Pseudo-inverses, such as the Moore-Penrose inverse $Cov^{\dagger}(F_n)$ (Penrose, 1955; Rao and Mitra, 1971) can of course be used; here and in the following, we indicate the resulting estimator as

$$\hat{\Sigma}_T^- = Cov^{\dagger}(F_n) \tag{3.1}$$

with the understanding that this reduces to the *traditional* "plug-in" estimator (T) $Cov^{-1}(F_n)$ whenever all eigenvalues of $Cov(F_n)$ are strictly positive. Unfortunately, although always defined, $\hat{\Sigma}_T^-$ can be highly unstable and carry large mean squared error in undersampled settings. This is due to the fact that, although the estimator disregards eigenvalues equal to 0, $Cov(F_n)$ is likely to have several strictly positive but very small eigenvalues (Raudys and Duin, 1998; Schäfer and Strimmer, 2005a).

As was noted in Chapter 1, many existing resampling based approaches, among which *bagging* (Breiman, 1996) and the *smoothed bootstrap* (Efron, 1979; Silverman and Young, 1987), can be used to "stabilize" estimators and improve their accuracy.

Resampling based approaches have been successfully applied to high dimensional genomic data analyses, including those requiring estimation of Σ^{-1}. For instance, Schäfer and Strimmer (2005a) used bagging (BB) to "stabilize" and reduce the mean squared error of Moore-Penrose inverses when estimating inverse variance-covariance (or correlation) matrices from microarray data – see also Sections 3.2 and 3.3.2. B resamples of size n are drawn from the data, the Moore-Penrose inverse of the variance-covariance matrix is computed on each such sample, and the resulting matrices are averaged to obtain the estimator

$$\hat{\Sigma}_{BB}^- = \frac{1}{B} \sum_{j=1}^{B} Cov^{\dagger}(F_{n(n)}^{(j)}) \tag{3.2}$$

In this chapter we investigate performance of the augmented bootstrap (AB) for the estimation of Σ^{-1}. As we have seen in Chapter 2, the augmented bootstrap employs the "noising rationale" underlying the smoothed bootstrap and, further-

more, borrows stabilization strength from bagging. This provides a straightforward and remarkably effective way of overcoming non-invertibility or instability in the inversion of the sample variance-covariance matrix when estimating Σ^{-1}.

Also non-resampling based approaches have been used to handle the estimation of the inverse variance-covariance (or correlation) matrix in high dimensional, undersampled settings. Schäfer and Strimmer (2005b) investigated *shrinkage* (SH) as a computationally parsimonious alternative to bagging Moore-Penrose inverses. Using a strictly positive definite $p \times p$ target matrix Ψ and a shrinkage intensity $\psi \in [0, 1]$, the estimator is defined as

$$\hat{\Sigma}^{-}_{SH} = [\psi\Psi + (1 - \psi)Cov(F_n)]^{-1} \tag{3.3}$$

Note that the matrix being inverted here is strictly positive definite even when $Cov(F_n)$ is not, provided $\psi > 0$. In practice, a class is selected for the target matrix (e.g. diagonal matrices with positive diagonal entries), and within such class Ψ is determined on the data estimating the quantities defining the class (e.g. setting the diagonal entries to be equal to the data coordinate variances). As will be discussed in Section 3.4.1, the augmented bootstrap method for estimating inverse variance-covariance matrices is analogous to a Monte-Carlo version of the ridge estimator (Foster, 1961; Hoerl, 1962), and therefore of the shrinkage estimator in equation 3.3. For the augmented bootstrap, an analog to the target matrix is the variance-covariance structure of the noise, which we take to be spherical. For shrinkage, the intensity ψ can be chosen arbitrarily from a range of values, or optimized on the data as a function of the target (as described in Schäfer and Strimmer, 2005b; Ledoit and Wolf, 2003). As remarked in Strimmer (2008), the shrinkage intensity is closely related to the smoothing variance of the augmented bootstrap. In the analyses by Schäfer and Strimmer (2005b), shrinking proved quite effective and robust to the choice of target class. Nevertheless, one can expect shrinkage to perform best in contexts where enough is known about the mechanisms generating the data to inform selection of the target class. Similar considerations apply for the augmented bootstrap and the choice of its noising variance-covariance structure. For instance, in the case of microarray data, a diagonal target (or the identity target when dealing with correlation matrices)

will often work very well, because the population-level variance-covariance matrix is likely to be sparse – i.e. have a preponderance of 0 off-diagonal entries; see Section 3.3.2.

In supervised dimension reduction applications, a recently proposed approach (Cook et al, 2007) allows one to bypass direct estimation of Σ^{-1} altogether. This is achieved by means of an iterative procedure that exploits the supervised nature of the problem, with a logic similar to that of Partial Least Squares Helland (1990) – some details are provided in Section 3.3.1.

As mentioned in Chapter 1, we consider $F_n(\tau^2) = F_n \circ N(0, \tau^2 I)$ in place of F_n, and thus use a spherical Gaussian noising, instead of reproducing the population variance-covariance structure (Σ) as in the traditional smoothed bootstrap described in Silverman and Young (1987). Sampling from $F_n(\tau^2)$ is implemented resampling with replacement from F_n and adding to each point an independent noise draw. However, since the noise is spherical, a generic sample $F_{n(h)}(\tau^2)$ of size h will have a non-singular variance-covariance matrix for any choice of $h > p$ and $\tau^2 > 0$, with probability 1, even when $n < p$ and therefore $Cov(F_n)$ is singular. In other words, the AB allows us to generate any number h of distinct and non-collinear points, and in particular a number in excess of the dimension p, even when the original data consists of $n < p$ points. For estimation of $\Sigma^{-1} = Cov^{-1}(F)$, the AB produces

$$\hat{\Sigma}^{-}_{AB} = Cov^{-1}(F_{n(mn)}(\tau^2)) \tag{3.4}$$

At finite sample sizes and *a fortiori* in undersampled settings, we expect the mean squared error of $\hat{\Sigma}^{-}_{AB}$ to first decrease and then increase as the smoothing variance τ^2 increases, and to be smaller than the mean squared error of $\hat{\Sigma}^{-}_{T}$ when τ^2 is small. The intuition is as follows: directions relevant to the population variance-covariance structure can be missing or almost missing in $Cov(F_n)$, especially when n is not large relative to p. A small amount of noise allows these directions to be "rescued" and contribute more stably to the inversion of the sample variance-covariance. However, as the smoothing variance increases, this advantage is overcome by the fact that noising alters the underlying variance-covariance structure towards sphericity. Note that this intuitive explanation is consistent with the theoretical derivations for functionals of the form 2.1 considered in Chapter 2. As we shall see through simulations in Section 3.2, $\hat{\Sigma}^{-}_{AB}$ does indeed show a behavior

consistent with this intuition, and outperforms $\hat{\Sigma}_T^-$ in a variety of scenarios.

Without loss of generality, the mn points forming $F_{n(mn)}(\tau^2)$ can be indexed as $x_{j,i}$, $j = 1, \ldots, m$, $i = 1, \ldots, n$, and thought of as m noised resamples of size n; $F_{n(n)}^{(j)}(\tau^2)$, $j = 1, \ldots, m$. Since the m mean vectors $\bar{x}_j = \frac{1}{n}\sum_{i=1}^n x_{j,i}$, $j = 1, \ldots, m$, and the overall mean vector $\bar{x} = \frac{1}{mn}\sum_{j=1}^m \sum_{i=1}^n x_{j,i}$ are estimates of the same quantity, the two matrices

$$\begin{aligned} Cov(F_{n(mn)}(\tau^2)) &= \frac{1}{mn}\sum_{j=1}^m \sum_{i=1}^n (x_{j,i} - \bar{x})(x_{j,i} - \bar{x})' \\ \overline{Cov}(F_{n(n)}(\tau^2)) &= \frac{1}{m}\sum_{j=1}^m Cov(F_{n(n)}^{(j)}(\tau^2)) \\ &= \frac{1}{m}\sum_{j=1}^m \frac{1}{n}\sum_{i=1}^n (x_{j,i} - \bar{x}_j)(x_{j,i} - \bar{x}_j)' \end{aligned}$$

are likely to show fairly similar behavior. This allows us to also interpret $\hat{\Sigma}_{AB}^-$ in (3.4), to some approximation, as the inverse of a bagged variance-covariance matrix, where bagging is implemented in a traditional sense, i.e. averaging over m bootstrap samples of size n. We therefore expect the mean squared error of the AB estimator to be a decreasing function of the sample size multiplier m, and this is indeed confirmed by our simulations in Section 3.2. Note though that here *bagging is combined with noising*. These observations confirm the notion that the augmented bootstrap can be viewed as a composition of smoothing and bagging, even though Σ^{-1} is not expressed explicitly in form 2.1.

Schäfer and Strimmer (2005a) did compare Moore-Penrose inversion of a bagged variance-covariance matrix to bagging the Moore-Penrose inverse (i.e. $\hat{\Sigma}_{BB}^-$ in (3.2)), and found the performance of the former inferior. However, as we shall see in Section 3.2, $\hat{\Sigma}_{AB}^-$ performs better than $\hat{\Sigma}_{BB}^-$ in a variety of scenarios, demonstrating the power of the "noising rationale" for tackling the estimation of Σ^{-1}.

3.2 Simulations

In this Section we use simulations to assess the performance of the AB estimator $\hat{\Sigma}_{AB}^-$ in (3.4) in comparison to: (i) the Moore-Penrose inverse of the sample

variance-covariance, $\hat{\Sigma}_T^-$ in (3.1) (i.e. the inverse neglecting the eigenspace corresponding to 0 eigenvalues; when $n > p$ and all eigenvalues of the sample variance-covariance are positive, this coincides with the standard inverse),(ii) the estimator based on bagging the Moore-Penrose inverse of the sample variance-covariance, $\hat{\Sigma}_{BB}^-$ in (3.2), and (iii) the estimator based on shrinking the sample variance-covariance prior to inversion, $\hat{\Sigma}_{SH}^-$ as implemented in R-package `corpcor`. This is a two-stage approach that separately shrinks variances towards their median, and covariances towards zero, with the shrinkage intensities optimally estimated from the data (Strimmer, 2008).

We fix the dimension of our space to be $p = 100$, and draw samples from $X \in \mathbb{R}^p$ normally distributed with mean $0 \in \mathbb{R}^p$ and three alternative variance-covariance matrices

$$\Sigma_1 = R_{(p,\rho)} \quad , \quad \Sigma_2 = \begin{pmatrix} R_{(0.5p,\rho)} & 0 \\ 0 & R_{(0.5p,\rho)} \end{pmatrix} \quad , \quad \Sigma_3 = \begin{pmatrix} R_{(0.1p,\rho)} & 0 \\ 0 & I \end{pmatrix} \tag{3.5}$$

where, for a generic dimension q and $\rho \in [0, 1]$

$$R_{(q,\rho)} \quad = \quad \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{pmatrix} \qquad q \times q$$

This choice allows us to simplify our analysis, eliminating scale (these are actually correlation matrices) and focusing on one parameter only (ρ). The spectral decomposition of $R_{(q,\rho)}$ is given by $(1 + (q-1)\rho)P_1 + (1-\rho)Q_1$, where P_1 and Q_1 are the orthogonal projection operators on the span of the unit vector and its orthogonal complement, respectively. For $0 < \rho \leq 1$, the eigenvalue $(1 + (q-1)\rho)$, which has multiplicity 1, dominates the eigenvalue $(1-\rho)$, which has multiplicity $q-1$. The difference between the two is $q\rho$, increasing with ρ; when $\rho = 1$, the second eigenvalue becomes 0 and the matrix collapses to rank 1. Another important feature of $R_{(q,\rho)}$ is that its inverse retains the same structure, and so do the

inverses of the three matrices considered for our simulations:

$$\Sigma_1^{-1} = R_{(p,\rho)}^{-1} \;,\; \Sigma_2^{-1} = \begin{pmatrix} R_{(0.5p,\rho)}^{-1} & 0 \\ 0 & R_{(0.5p,\rho)}^{-1} \end{pmatrix} \;,\; \Sigma_3^{-1} = \begin{pmatrix} R_{(0.1p,\rho)}^{-1} & 0 \\ 0 & I \end{pmatrix}$$

$$R_{(q,\rho)}^{-1} = \begin{pmatrix} \frac{(q-2)\rho+1}{(1-\rho)(1+(q-1)\rho)} & -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} & \cdots & -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} \\ -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} & \frac{(q-2)\rho+1}{(1-\rho)(1+(q-1)\rho)} & \cdots & -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} & -\frac{\rho}{(1-\rho)(1+(q-1)\rho)} & \cdots & \frac{(q-2)\rho+1}{(1-\rho)(1+(q-1)\rho)} \end{pmatrix}$$

This implies, among other things, that the correlation and partial correlation structures among the coordinates of X are equivalent in terms of placement of zeros (more details on partial correlations are given in Section 3.3.2). We therefore have the following: (1) Σ_1 has no 0 entries, and the resulting simulated data concentrate around one direction as ρ grows. (2) Σ_2 has $0.5p^2$ zero entries, and represent a scenario in which the variables are divided in two groups of equal size, correlated (and partially correlated) within each group and uncorrelated between the groups, with strength of the grouping, as well as concentration of the data around one direction within each group, increasing with ρ. (3) Σ_3 is sparse, with $(0.99p - 0.9)p$ zero entries, and represents a scenario with a small group $(0.1p)$ of correlated (and partially correlated) variables among many $(0.9p)$ uncorrelated ones. The strength of the association in the small group increases with ρ, but even at high ρ the data does not tend towards low dimensionality (it tends to concentrate around $0.9p + 1$ directions).

For sample size, we consider $n_1 = 10p$ (a scenario where the sample size is reasonably large relative to the dimension), $n_2 = 1.1p$ (a scenario in which the sample variance-covariance matrix is still invertible, but a very poor estimator of its population counterpart) and $n_3 = 0.5p$ (a scenario in which the sample size falls far short of the dimension).

To assess the performance of an estimator in a given simulation setting (as identified by a sample size and a variance-covariance matrix), we compute mean and standard deviation of its *relative squared error* (RSE) over 50 independent

replications. The RSE is defined as

$$RSE(\hat{\Sigma}^{-}; \Sigma^{-1}) = \frac{||\hat{\Sigma}^{-} - \Sigma^{-1}||^2}{||\Sigma^{-1}||^2} \tag{3.6}$$

where the square norm of a generic $q \times q$ matrix A is given by $||A||^2 = \sum_{i,j=1}^{q} a_{i,j}^2$ (Frobenius norm).

Table 3.1 contains results for nine simulation settings corresponding to n_j, $j = 1, 2, 3$ and Σ_j, $j = 1, 2, 3$ with $\rho = 0.5$. Notably, the performance of the T and BB estimators does *not* depend on the underlying variance-covariance matrix, while the performance of the SH estimator does. More specifically, SH does better the sparser the variance-covariance, and is the best performing estimator for Σ_3 – which can be explained based on its reliance on sparsity. However, the AB estimator is substantially better than all other estimators for Σ_1 and Σ_2 at all sample sizes, including samples with $n_1 = 10p$, where BB does worst, SH is very close to T, and AB essentially halves the mean RSEs of SH and T. For Σ_3, AB is still substantially better than T and BB at all sample sizes, and achieves a performance similar to that of SH for $n_1 = 10p$. Surprisingly, although the AB is tightly connected with shrinkage for variance-covariance matrix estimation, its performance does not seem to depend on the variance-covariance structure of the data. We attribute this observation to the different optimization mechanisms for the shrinkage intensities and smoothing variance, with the former being estimated from the data, and the latter optimized on a range of values (see below).

In terms of dependence on sample size, T markedly shows the *resonance* phenomenon described in Raudys and Duin (1998) and Schäfer and Strimmer (2005a), i.e. a peak of the mean RSE at $n \approx p$, when inversion is made highly unstable by the inclusion of positive but very small eigenvalues. This phenomenon is greatly mitigated by bagging, although its "shadow" remains in the mean RSE of BB, which still increases by about one order of magnitude at $n_2 = 1.1p$ relative to $n_1 = 10p$ and $n_3 = 0.5p$. Interestingly, also SH shows an increase in mean RSE at $n_2 = 1.1p$ for Σ_1 and Σ_2, perhaps because the choice of shrinking intensities in these settings mitigates but does not completely eliminate the potential for instability in their inversion. Remarkably, AB shows no sign of resonance; the deterioration in its performance when passing from $n_1 = 10p$ to $n_2 = 1.1p$ is much smaller than

that of all other estimators for Σ_1 and Σ_2, and of that of T and BB for Σ_3, and the mean RSE of AB remains practically unchanged when passing from $n_2 = 1.1p$ to $n_3 = 0.5p$.

n	T	SH	BB	AB
	Σ_1			
$10p$	0.151 (0.005)	0.143 (0.004)	0.241 (0.009)	0.070 (0.001)
$1.1p$	2417.72 (1762.05)	11.629 (1.933)	2.063 (0.227)	0.186 (0.002)
$0.5p$	0.994 (0.107)	9.942 (3.622)	0.723 (0.007)	0.194 (0.002)
	Σ_2			
$10p$	0.148 (0.005)	0.135 (0.005)	0.237 (0.008)	0.069 (0.001)
$1.1p$	3030.24 (1815.57)	4.230 (0.456)	2.086 (0.204)	0.187 (0.003)
$0.5p$	0.969 (0.080)	2.598 (0.602)	0.731 (0.006)	0.192 (0.002)
	Σ_3			
$10p$	0.141 (0.005)	0.056 (0.001)	0.229 (0.007)	0.075 (0.001)
$1.1p$	2695.31 (1850.81)	0.079 (0.004)	1.872 (0.222)	0.219 (0.003)
$0.5p$	0.964 (0.079)	0.079 (0.005)	0.755 (0.006)	0.234 (0.003)

Table 3.1. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.5$.

The same simulation scenarios were run also for $\rho_1 = 0.1$ and $\rho_3 = 0.9$. Results were consistent with the above description (data presented in Appendix B). Only SH differed notably in performance as a function of ρ, behaving better at smaller ρ's – i.e. when each of the variance-covariance matrices, compatibly with their structures, become closer to the diagonal target. Also at small and high ρ's though, AB was best for Σ_1 and Σ_2, and better than T and BB for Σ_3, at all sample sizes, showing no sign of resonance. Once again, substantial differences between AB and SH performance are likely a consequence of the different values of the tuning parameters used for these methods.

As a means to assess the performance of the AB estimator of the inverse variance-covariance matrix in comparison to the T, SH and BB estimators in cases where the data is non-Gaussian, we took data generated from $X \sim N_{100}(0, R_{(p,\rho)})$ with $p = 100$ and $\rho = 0.5$, and squared each of the coordinates. This results in data from $\tilde{X}$ whose coordinates all have χ^2 distribution with 1 degree of freedom, and whose variance-covariance matrix, which can be derived based on the fourth

mixed moments of the underlying Gaussians (Kamat, 1981) is

$$\tilde{\Sigma} = \begin{pmatrix} 2 & 2\rho^2 & \dots & 2\rho^2 \\ 2\rho^2 & 2 & \dots & 2\rho^2 \\ \vdots & \vdots & \ddots & \vdots \\ 2\rho^2 & 2\rho^2 & \dots & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0.5 & \dots & 0.5 \\ 0.5 & 2 & \dots & 0.5 \\ \vdots & \vdots & \ddots & \vdots \\ 0.5 & 0.5 & \dots & 2 \end{pmatrix} \tag{3.7}$$

Table 3.2 contains results for sample sizes n_j, $j = 1, 2, 3$, in the same format as the Gaussian simulation results in Table 3.1. Notably, the AB estimator performs best at all sample sizes also in this comparison, and appears to be far less affected than the other estimators by non-Gaussianity – the performance of AB is only marginally worse than, say, for the Gaussian simulations with Σ_1 and $\rho = 0.5$, while the performance of all other estimators deteriorates substantially.

n	T	SH	BB	AB
$10p$	0.336 (0.037)	0.240 (0.025)	0.568 (0.064)	0.094 (0.005)
$1.1p$	11114.3 (11514)	0.782 (0.231)	12.616 (2.230)	0.171 (0.010)
$0.5p$	3.031 (0.915)	0.328 (0.155)	0.657 (0.017)	0.169 (0.011)

Table 3.2. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n. Non-Gaussian simulation data; $p = 100$ and $\rho = 0.5$.

For implementing the SH method, we used code from the R library `corpcor` created by Schäfer and Strimmer (2005b) and available through CRAN (http://cran.r-project.org/), with the default diagonal target and optimal shrinkage parameters endogenously evaluated by the routine. For the BB method, the only tuning parameter to be fixed is the number of resamples used in bagging; we used 70; this value is lower than the ones used by Schäfer and Strimmer in applications (Schäfer and Strimmer, 2005a,b), but is such that BB is based on the same overall "amount of resampling" (a total number of $70n$ points in each simulation run) as the AB method – see below.

For the AB method, Table 3.1 and 3.2 report mean RSE and standard deviation for sample size multiplier fixed at $m = 70$, and smoothing variance τ^2 optimized on a grid for each simulation setting – the grid extends from 0.01 to 1.2 for the Gaussian simulations with Σ_j, $j = 1, 2, 3$ (where the variance of all coordinates is 1), and from 0.01 to 2.5 for the χ^2 simulations with $\tilde{\Sigma}$ (where the variance of all

coordinates is 2).

Figure 3.1 illustrates the behavior of the mean RSE for AB against m (for optimal values of τ^2; panel A) and against τ^2 (for $m = 70$; panel B). These curves were computed from the Gaussian simulation data with Σ_1 and $\rho = 0.5$, but the behavior is similar for other simulation settings. The mean RSE decreases monotonically as a function of the sample size multiplier, but it quickly "levels off"; for all sample sizes under consideration ($n_1 = 10p = 1000$, $n_2 = 1.1p = 110$ and $n_3 = 0.5p = 50$) increasing m beyond 50 induces little improvement. On the other hand, the mean RSE decreases to a minimum and then increases again as the smoothing variance increases, consistent with the intuition outlined in Section 3.1. Importantly, the minimum of the mean RSE occurs at larger values of τ^2 the

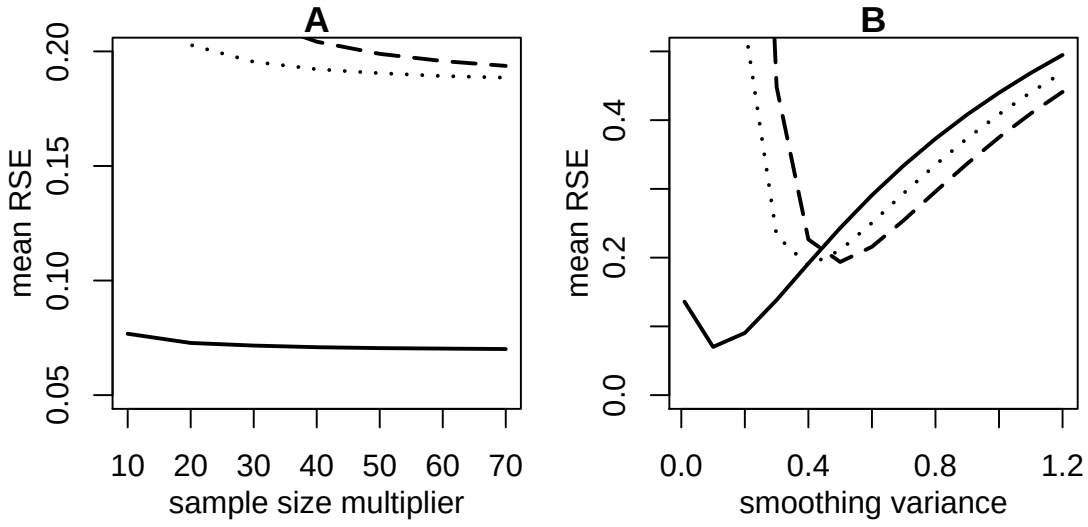

Figure 3.1. Plots of the mean RSE for AB against the sample size multiplier m (with τ^2 fixed at its optimal values; panel A), against the smoothing variance τ^2 (with $m = 70$; panel B). The data were generated from a Gaussian distribution using Σ_1 and $\rho = 0.5$. Different lines correspond to different sample sizes: *solid* for $n_1 = 10p = 1000$, *dotted* for $n_2 = 1.1p = 110$, and *dashed* for $n_3 = 0.5p = 50$. Corresponding to these sample sizes, the values of mean RSE for T are 0.151, 2417.72 and 0.994, respectively.

smaller the sample size. Location of the minimum also changes depending on the data (e.g. it increases with sparsity of the underlying variance-covariance matrix, and with the values of the coordinate variances – see Table 3.3).

While we do not explicitly represent Σ^{-1} in form 2.1, our observations generally agree with the theoretical developments in Chapter 2. The MSE considered as a

n	Σ_1	Σ_2	Σ_3	$\tilde{\Sigma}$
$10p$	0.1	0.1	0.2	0.5
$1.1p$	0.4	0.4	0.7	1.3
$0.5p$	0.5	0.5	0.9	1.5

Table 3.3. Optimal smoothing variance τ^2 of the AB method for various Gaussian (columns 1–3) and non-Gaussian (column 4) simulation scenarios, and choices of n; $p = 100$ and $\rho = 0.5$.

function of the smoothing variance τ^2 has a "V"-shape. As for the choice of the smoothing variance, we advise to consider several values of τ^2, in a range determined as a function of the observed sample coordinate variances (e.g. extending past a given quantile of such variances; note that in Table 3.3 the optimal τ^2 is always smaller than the common underlying value of the coordinate variances). If the application allows one to define an objective function (e.g. a cross-validation performance measurement for supervised problems; see Section 3.3.1) an optimization can be performed on a grid covering this range, based again on computing feasibility.

On the other hand, the sample size multiplier does not affect performance substantially, provided it is "large enough". Recall that for functionals of the form 2.1 the mean squared error of the AB estimator is a quadratic function of the reciprocal of the sample size multiplier g (see Proposition 2.2.1); and after transformation from m to g in terms of the first part of the Proposition 2.2.2, the behavior observed in Figure 3.1 appears consistent with the case in which the unique root of the first derivative of the $MSE(\theta^{AB}(F_n); g)$ is non-positive. In terms of the coefficients used in Chapter 2, this happens when $\beta_{bias}^{(bag)} + \tau^2 \delta_{bias}^{(inter)} + \tau^4 \gamma_{bias}^{inter} > 0$ and $\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)} + \tau^2(\alpha_{bias}^{(inter)} + \alpha_{var}^{(inter)}) + \tau^4(\beta_{bias}^{(inter)} + \beta_{var}^{(inter)}) \geq 0$. Given the size of the data and the available computing power, therefore, for applications that involve estimating the inverse of the variance-covariance matrix, it is sensible to fix a sample size multiplier m that is large but not too burdensome in terms of computing. Some comments on computing requirements are given in Section 3.5.

3.3 Applications to genomic data

3.3.1 DNA sequence data: supervised dimension reduction and classification based on motif frequencies

Here, we revisit the motivating example from Chapter 1 and use part of the data analyzed in Carrel et al (2006). We restrict attention to the frequencies of $p = 248$ DNA meta-motifs of length 12 (meta-12-mers; see Section 1.2 for details) that were pre-selected as showing significant over-representation in windows of size 200 Kb (thousands of bases) around the start sites of inactivated (Su) vs non-inactivated (Es) genes. Again, we consider two data sets: one comprises $n = 63 < p$ genes with experimentally validated Su/Es status (42 Su and 21 Es genes) almost all of which are located on Xp22, the other comprises $n = 365 > p$ genes with validated Su/Es status from the whole chromosome X (318 Su and 47 Es genes). For this larger set, the number of Es genes is much smaller relative to the total, and although p does not exceed n, we still have only 1.47 observations (genes) per variable (meta-12-mer frequency).

For both data sets, the aim is to perform supervised dimension reduction, estimating the direction β in the 248-dimensional meta-12-mer frequency space that is most relevant for the 2-way classification of genes as Su or Es. To do this, we consider a *Sliced Inverse Regression* approach (SIR, Li, 1991). SIR can be used with both categorical and continuous responses – in the latter case, the response is "sliced" into discrete classes, from which the name. With a 2-way classification, SIR works in a way very similar to classic Linear Discriminant Analysis (LDA, Hastie et al, 2001). The direction aligned with the difference of the two class means in the predictors space, i.e. the span of the vector $\nu = \frac{(\mu_{Su} - \mu_{Es})}{||\mu_{Su} - \mu_{Es}||}$, is "benchmarked" against a reference variability structure, which is given by the average within-class variance-covariance matrix in the case of LDA, and by the overall variance-covariance matrix Σ in the case of SIR. Technically, benchmarking means repositioning ν by pre-multiplication with Σ^{-1}. Hence, the population object targeted by SIR is

$$\beta = \Sigma^{-1}\nu \tag{3.8}$$

where ν is straightforwardly estimated using the sample class means, and the issue

for undersampled settings such as the one in this example is how to estimate Σ^{-1}, or possibly how to estimate β without having to estimate Σ^{-1}. Once an estimate $\hat{\beta}$ is produced, $\ell = \hat{\beta}'x$ can be used as a classification score. In particular, we apply a simple "naïve Bayes" classifier, which attributes a generic gene with 12-mer frequency vector x to Su or Es based on whether its score $\ell = \hat{\beta}'x$ is above or below a threshold λ tuned for optimal discrimination (see below).

Here we compare the direction estimated with the AB method, $\hat{\beta}_{AB} = \hat{\Sigma}_{AB}^{-}\hat{\nu}$, the one estimated with the traditional approach, which uses the Moore-Penrose inverse of the sample variance-covariance matrix, $\hat{\beta}_T = \hat{\Sigma}_T^{-}\hat{\nu}$, and the one estimated with a recently proposed method by Cook, Li, and Chiaromonte (2007), $\hat{\beta}_{CLC} = \hat{C}\hat{\nu}$. In the latter, estimation of Σ^{-1} is bypassed altogether, using an iterative approach similar to Partial Least Squares (Helland, 1990). More precisely, the matrix used to reposition $\hat{\nu}$ is

$$\hat{C} = \hat{U}_u(\hat{U}_u' Cov(F_n)\hat{U}_u)^{-1}\hat{U}_u' \tag{3.9}$$

where $\hat{U}_u = (\hat{\nu}, Cov(F_n)\hat{\nu}, \ldots, Cov^{u-1}(F_n)\hat{\nu})$. Thus, CLC requires taking powers (instead of the inverse) of the sample variance-covariance matrix, along with inversion of the $u \times u$ matrix $\hat{U}_u' Cov(F_n)\hat{U}_u$. With parsimonious choices of the tuning parameter u, this provides a very effective supervised dimension reduction tool for undersampled settings – the properties of $\hat{\beta}_{CLC}$ as an estimator of β are discussed in detail in Cook et al (2007). Of note here is the fact that CLC, unlike AB, or the traditional approach, is tailored directly to supervised dimension reduction and does not produce an estimate of Σ^{-1}.

Table 3.4 reports leave-one-out cross-validation success rates (overall, and separately for Su and Es genes) achieved with T, CLC and AB on the Xp22 and whole-X data sets. The CLC tuning parameter is fixed at $u = 20$; for both data sets, this value should guarantee effectiveness of the method according to prescriptions in Cook et al (2007) based on the eigenvalues of the sample variance-covariance matrix. The AB sample size multiplier is fixed at $m = p = 248$ for the Xp22 data set (where $n = 63$), and at a computationally parsimonious $m = 50$ for the whole-X data set (where such a value is reasonable because $n = 365$). For each data set, the AB smoothing variance is optimized on a grid extending well past the 90th

percentiles of the sample coordinate variances, resulting in $\tau^2 = 2.7$ for the Xp22 data ($n = 63$) and $\tau^2 = 0.3$ for the whole-X data ($n = 365$). The classification threshold λ is also optimized on a grid, separately for each data set and method. For both τ^2 and λ, optimization is pursued maximizing the sum of Es and Su cross-validation success rates (correctly attributed Es's over total number of Es's, plus correctly attributed Su's over total number of Su's). This is preferred to maximizing the overall success rate (correctly attributed genes over total number of genes) because the number of Es and Su genes in the data is very different, and the biological import of correctly predicting escaping and not escaping inactivation quite separate.

Success Rate	Xp22			whole-X		
	T	CLC	AB	T	CLC	AB
Overall	83%	95%	95%	89%	89%	91%
Es	100%	90%	90%	79%	87%	96%
Su	74%	98%	98%	91%	88%	90%

Table 3.4. Leave-one-out cross-validation success rates for various methods, X-inactivation data.

For the Xp22 data set ($n = 63$), AB and CLC have identical and excellent performance, and substantially outperform T both in terms of overall success rate (95% for AB and CLC vs 83% for T) and in terms of sum of Es and Su success rates (188% for AB and CLC vs 174% for T). For the whole-X data set ($n = 365$) the three estimators have similar overall success rates (89-91%) but AB substantially outperforms CLC, which in turn does better than T, in terms of sum of Es and Su success rates (186% for AB, 175% for CLC, 170% for T).

3.3.2 Microarray data: Gene interactions and networks based on transcription profiles

The yeast *Saccharomyces Cervisiae*, as one of the simplest and best studied eukaryotic organisms, has long been a model of choice for investigating pathways and networks through which genes and their products interact. The potential of these investigations has been revolutionized by the sequencing of the yeast genome

(Cherry et al, 1997) and the advent of microarrays (Lashkari et al, 1997), which allow researchers to record the levels of transcription (i.e. the abundances of mRNA) for all known and putative yeast genes simultaneously.

Microarrays are used to trace transcription profiles for thousands of genes across time courses and/or treatment conditions, and covariation (or countervariation) among profiles is taken as evidence of genes partaking in functional modules, i.e. performing similar or related functions, and possibly being activated or suppressed by common control mechanisms. Integrating this information with information on the location and status of sequences in the genome that regulate gene transcription, and with data on interactions among proteins, one can start piecing together a picture of the underlying interaction networks genome-wide (see for instance Lee et al, 2002, for yeast, and Gunsalus et al, 2005, in *C. Elegans*, another model organism).

Here we consider a subset of the microarray data analyzed in Gasch et al (2000) to investigate the transcriptional response of yeast to various environmental stresses, pre-processed for normalization and imputation of missing values. We restrict attention to $p = 321$ yeast genes (out of $6,141$) selected as having a coefficient of variation (ratio between standard deviation and absolute mean value) larger than 10 across $n = 10$ time points sampled throughout the so-called *stationary phase*. This is a developmental state entered by yeast cells running out of nutrients in their environment, and is characterized by various physiological responses coordinated with a cell cycle arrest (Werner-Washburne et al, 1993). Cells in stationary phase do not proliferate, but remain viable and resume growth when nutrients become available again.

Gaussian Graphical Models (Whittaker, 1990) have been broadly used in recent literature to produce network inferences based on covariation in gene transcription levels (Schäfer and Strimmer, 2005b and references therein; Lezon et al, 2006). In these models, associations among genes (edges in an undirected graph) are assessed using *partial correlation coefficients*; that is, correlations between each pair of genes conditional to all other genes included in the analysis (if the data were indeed drawn from a multivariate Gaussian, non-significant partial correlations, i.e. missing edges, would represent conditional independence). The $p \times p$ matrix

of partial correlation coefficients is given by

$$\Pi = \left(-\frac{\omega_{k,j}}{(\omega_{k,k}\omega_{j,j})^{1/2}} \right) \tag{3.10}$$

where $\omega_{k,j}$, $k, j = 1, \ldots, p$ are the entries of the inverse correlation matrix

$$R^{-1} \quad = (D(\sigma_{j,j}^{-1/2})\Sigma D(\sigma_{j,j}^{-1/2}))^{-1} \quad = D(\sigma_{j,j}^{1/2})\Sigma^{-1} D(\sigma_{j,j}^{1/2})$$

and the issue is again how to estimate Σ^{-1} from undersampled data. We compare the partial correlations estimated with the AB approach, $\hat{\Pi}_{AB}$, the bagging approach, $\hat{\Pi}_{BB}$, and the shrinkage approach, $\hat{\Pi}_{SH}$. BB was run with 1000 resamples, and SH was run directly on the correlation matrix, using the identity matrix as target and with endogenously evaluated shrinkage intensities, as implemented in R-package `corpcor` (Schäfer and Strimmer, 2005b). Since in this application the sample size is very small ($n = 10$ for $p = 321$, as compared to, say $n = 63$ for $p = 248$ in Xp22 data set of Section 3.3.1), the multiplier for AB was fixed at $m = 1000$ – this provides $n \cdot m/p = 10 \cdot 1000/321 = 31$ data points per variable, comparable to the number used in our "$n < p$" simulation scenarios (where $n \cdot m/p = 50 \cdot 70/100 = 35$). Unlike in the simulations of Section 3.2, and in the X-inactivation application of Section 3.3.1, here there is no straightforward objective function relative to which the AB smoothing variance can be optimized; the results discussed below correspond to $\tau^2 = 0.5$.

Figure 3.2A presents the distributions of partial correlation coefficients generated by the three methods. The BB partial correlations are very spread, while those produced by SH are concentrated around 0 and extend in very thin tails. This is a desirable feature, since the expectation is that the underlying gene network will comprise only a very small share of all possible pair-wise associations. The AB partial correlations have a distribution much closer to that of SH than to that of BB, although with thicker tails. Figure 3.2B presents a scatter-plot of the SH and AB partial correlations. While there appears to be good concordance between the two methods, interestingly, the nine gene pairs with strongest AB associations (corresponding to black points in the plot) show relatively weak SH associations. In other words, while by and large AB (with a fixed value of

the smoothing variance) "scales up" association signals rendered by SH (with the shrinkage intensities optimized on the observed data), it also appears to be able to capture *different* signals. The nine gene pairs with strongest AB partial correlations do not correspond to interactions annotated in the "Saccharomyces Genome Database" (http://www.yeastgenome.org/), and as in most of these analyses, it is impossible to ascertain whether these associations are biologically meaningful without an appropriately designed experimental follow-up. But from a statistical perspective, these represent very sizable marginal associations (correlation coefficients as extreme as -0.77 and $+0.48$) which are to a large extent "absorbed" by conditioning when SH with optimal shrinkage intensities is used to perform the required inverse estimate, and not when AB is used.

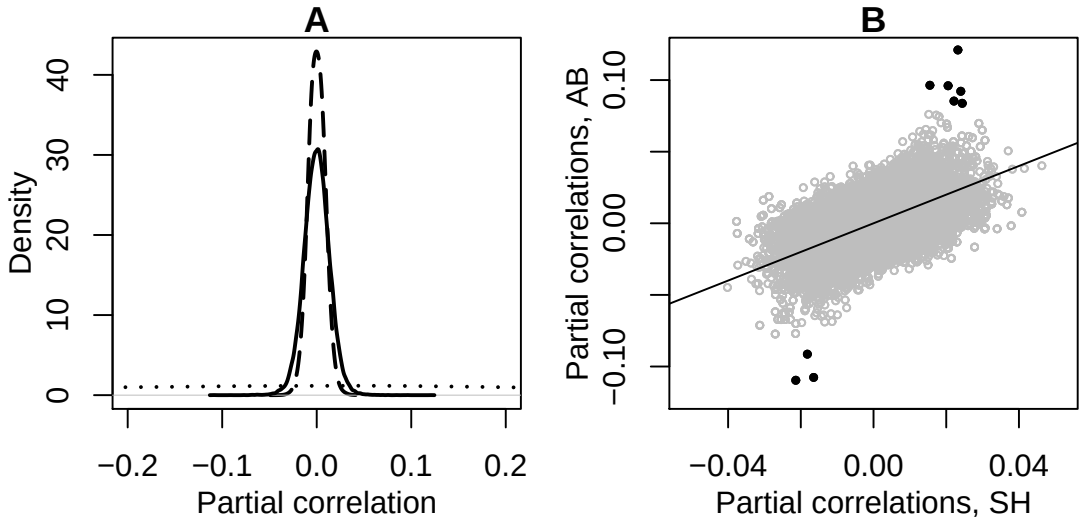

Figure 3.2. Panel A: smoothed distributions of partial correlation coefficients obtained with the AB (*solid* line), SH (*dashed* line) and BB (*dotted* line) methods. Panel B: scatterplot of partial correlation coefficients obtained with AB against those obtained with SH. The 1:1 diagonal is superimposed for visualization purposes. Gene pairs with extreme AB partial correlations but moderate SH partial correlations are shown with solid black circles.

Next, for each method, we form the degree distribution resulting from the 2056 (2%) partial correlations with most extreme values (i.e. we take each of the 321 genes, count how many of the coefficients between it and all other genes fall in the top 1% and bottom 1% of the partial correlations distribution, and create the distribution of these counts). This way, we fix the "level of sparsity" (number

of associations) and observe whether the methods differ in how associations are distributed – the expectation being that gene networks are *small world*-like and characterized by *hubs* (genes to which many genes are connected). The three degree distributions are presented in Figure 3.3. The BB degrees present a very strong mode at $6-7$, and do not have a substantial skewness or extended tail to the right – approximating what one would expect if associations were distributed "at random" among genes. Thus, in addition to not being effective for "pruning", BB also appears not to generate hubs with its strongest associations. At the opposite extreme, the SH degrees are much closer to the power law distributions of small world networks – with a weaker mode at $0-1$, and a thick right tail. Also here, due to the different values of the tuning parameters, AB behaves as a slightly more permissive version of SH. The AB degrees have a distribution more similar to that of SH than to that of BB, with a mode at $1-2$, and a substantial tail to the right. Moreover, AB appears to identify *bigger* and *different* hubs than SH, with higher and more outlying degrees in the right tail of the distribution. The genes EFT1, CAT5 and SPE2 have the largest AB degrees (57, 61 and 71, respectively), and relatively low SH degrees (12, 22 and 5). Interestingly, the same three genes also appear as members of six out of the nine pairs with strongest AB and relatively weak SH partial correlations discussed above (EFT1 in 2 pairs, CAT5 in 2 pairs, and SPE2 in 3 pairs; EFT1 and CAT5 actually constitute one such pair). Once again, existing annotations of gene interactions do not allow us to ascertain whether these genes may represent biologically meaningful hubs during the yeast stationary phase. But from a statistical perspective, they illustrate the marked differences in results that can be generated by employing related methods with the different amounts of smoothing.

3.4 Discussion and modifications

3.4.1 Why *simulating* a sphere?

As far as variance-covariance matrices are concerned, at the population level we can describe convolution with the Gaussian noise in the following way: if $X \sim F$, $\varepsilon \sim N(0, \tau^2 I)$ and X is independent of ε, then the noised $F \circ \epsilon_{\tau^2}$ is the distribution

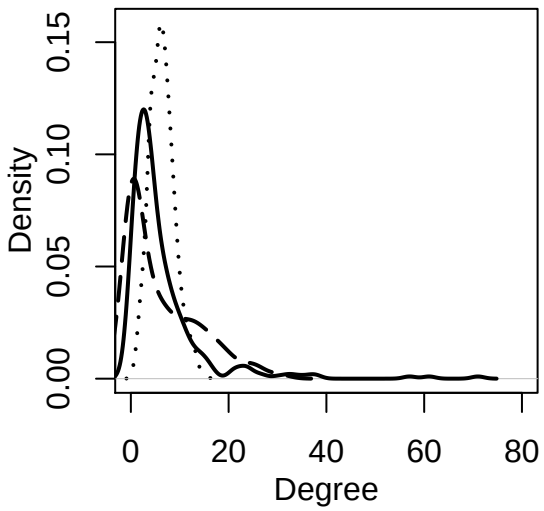

Figure 3.3. Smoothed degree distributions for the networks resulting from the top 2% partial correlations obtained with the AB (*solid* line), SH (*dashed* line) and BB (*dotted* line) methods.

of

$$X^* = X + \varepsilon$$

and

$$\begin{aligned} \Sigma_{X^*} &= Cov(X^*) = Cov(X + \varepsilon) \\ &= Cov(X) + Cov(\varepsilon) + 2cov(X, \varepsilon) \\ &= Cov(X) + Cov(\varepsilon) = \Sigma_X + \tau^2 I \end{aligned}$$

Therefore, noising the data by convolution with a spherical Gaussian and applying a ridge-type regularization (i.e. adding a multiple of the identity matrix) (Foster, 1961; Hoerl, 1962) are, in expectation, equivalent operations (Strimmer, 2008; Schäfer, 2008; Keleş and Chun, 2008). Consequently, their inverses should have similar behaviors.

Because of this connection, the AB estimator $\hat{\Sigma}^-_{AB}$ proposed in equation 3.4 can be seen as a simulated analogue of the *ridge regularized estimator* $(Cov(F_n) + \tau^2 I)^{-1}$ (Strimmer, 2008; Keleş and Chun, 2008). Note also that the ridge regularized estimator is proportional to the shrinkage estimator 3.3 with the identity

target matrix, because

$$Cov(F_n) + \tau^2 I \propto (1-\psi)\left(Cov(F_n) + \frac{\psi}{1-\psi} I\right).$$

Taking into explicit account the fact that the AB resamples from the data multiple times, the closest parallel is in fact with a *bagged regularized estimator* (Schäfer, 2008):

$$\left(\frac{1}{m}\sum_{j=1}^{m} Cov(F^{(j)}_{n(n)}) + \tau^2 I\right)^{-1}$$

where $F^{(j)}_{n(n)}$ indicates the empirical distribution of a bootstrap sample of size n drawn from F_n. In Section 3.1, we noted that for a variance-covariance matrix the AB approach would mimick bagging with noise; the insight here is that one might combine a bagging with regularization instead, sparing the nonnegligible computational cost of producing the noise draws. Another take is to bag after inversion to obtain an *outer bagged regularized estimator* (Keleş and Chun, 2008):

$$\frac{1}{m}\sum_{j=1}^{m}\left(Cov(F^{(j)}_{n(n)}) + \tau^2 I\right)^{-1}.$$

Using one of the simulation scenarios described in Section 3.2 (data from a $p = 100$ dimensional Gaussian distribution, with the variance-covariance matrix Σ_1 in Equation 3.5, $\rho = 0.5$, and the three sample sizes $n_1 = 1000$, $n_2 = 110$, $n_3 = 50$), we compared the regularized, bagged regularized, AB and outer bagged regularized estimators, all computed with the *same* smoothing variance τ^2 (selected on a grid as to optimize the performance of AB; see Table 3.3). The number of bootstrap samples for the two bagged estimators and the sample size multiplier for the AB were also the same ($m = 70$). Averages and standard deviations of the relative squared errors for the four estimators over 50 simulation runs are reported in Table 3.5 (this reproduces in part results from Table 1 in Keleş and Chun (2008)). The regularized, bagged regularized and AB estimators present practically undistinguishable performances. Notably, bagging (or multiplying the sample size) does not produce sizeable gains in performance – suggesting that for these data regularization plays the lion share in terms of variance-covariance stabilization prior to

the inversion. However, the outer bagging performs substantially better, perhaps because even with the regularized variance-covariance, there is still room to stabilize the inversion – this is consistent with comparisons between bagging prior of after Moore-Penrose inversion in Schäfer and Strimmer (2005a).

n	Reg	Bag Reg	AB	Out Bag Reg
$10p$	0.069 (0.001)	0.067 (0.001)	0.070 (0.001)	0.069 (0.001)
$1.1p$	0.186 (0.003)	0.184 (0.003)	0.187 (0.003)	0.117 (0.002)
$0.5p$	0.185 (0.003)	0.187 (0.003)	0.192 (0.003)	0.101 (0.002)

Table 3.5. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n. Gaussian simulation data; $p = 100$ and Σ_1, with $\rho = 0.5$.

Similar results could be observed under all the simulation scenarios used in Section 3.2 (see Appendix B for additional tables); regardless of the degree of sparsity in the underlying Σ, and even when the data is non-Gaussian, the AB estimator of Σ^{-1} behaves as the bagged regularized estimator, and thus as a simulated analogue of the regularized estimator.

The above observations allow us to view AB as a *computational stabilization* approach that combines a data *multiplication component*, akin to bagging, with a *regularization component* implemented by noising. The approach is fully general, and can be employed to mitigate the adverse effects of data shortage for all sorts of estimation problems and statistical procedures, whether or not they involve covariances and their inverses. We argue this point further with an application to classification trees in Chapter 4 below.

3.4.2 ... And why a *sphere*?

An interesting modification to the augmented bootsrap scheme was proposed in Li (2008). Recall, that instability in inverting the sample variance-covariance matrix, and the need to forego some directions in the inversion, are due to the presence of very small or zero eigenvalues. If, instead of noising spherically, one added noise only along the eigendirections of $Cov(F_n)$ with variance below a given threshold, the method would not act upon the dominant variability structure of the data, while still achieving the required stabilization. In particular, noising could be

used to raise all variances (eigenvalues) below τ^2 to τ^2 itself. Formally this can be described as follows. Let the spectral decomposition of the sample variance-covariance matrix be:

$$\hat{\Sigma} = \sum_{i=1}^{p} \hat{\lambda}_i \hat{v}_i \hat{v}_i^T.$$

Then noise draws could come from $N(0, \hat{\Sigma}_\epsilon)$, where

$$\hat{\Sigma}_\epsilon = \sum_{i=1}^{p} f(\lambda_i) \hat{v}_i \hat{v}_i^T,$$

and

$$f(\lambda) = \begin{cases} 0 & if\ \lambda > \tau^2 \\ \tau^2 - \lambda & if\ \lambda \leq \tau^2 \end{cases}$$

(see Li (2008) for details). This intuition indeed leads to substantial gains in performance, as can be seen in column 1 of Table 3.6 (this reproduces in part results from Table 1 in Li (2008)).

n	Adpt AB	Out Bag Adpt	Modified Bag Reg
$10p$	0.026 (0.001)	0.025 (0.001)	0.011 (3×10^{-5})
$1.1p$	0.095 (0.003)	0.056 (0.001)	0.012 (2×10^{-4})
$0.5p$	0.127 (0.003)	0.169 (0.351)	0.011 (0.005)

Table 3.6. Mean (and standard deviation) of the Relative Squared Error for adaptive augmented bootstrap, outer bagged regularized, and modified bagged regularized estimators of Σ^{-1} with the adaptive target. Various choices of n. Gaussian simulation data; $p = 100$ and Σ_1, with $\rho = 0.5$.

The notion of data-adaptive target choice provides an attractive compromise between spherical noising (or ridge regularization) and noising that preserves the observed variance-covariance structure of the sample, as prescribed by the original smoothed bootstrap procedure (Silverman and Young, 1987).

In light of the connections drawn in Section 3.4.1 above, although $\hat{\Sigma}_\epsilon$ is not a full rank matrix, B. Li's proposal shows us how to introduce an effective data-adaptive element to the augmented bootstrap – whether the aim is to stabilize estimation

of an inverse variance-covariance, or any other statistical procedure. In fact, $\hat{\Sigma}_\epsilon$ can be used in place of $\tau^2 I$ to create "adaptive" variants also for the regularized and bagged regularized estimators, with results practically indistinguishable from those in column 1 of Table 3.6. Interestingly, the data-adaptive $\hat{\Sigma}_\epsilon$ affords gains even when the data is scarce ($n = 1.1p$ and $n = 0.5p$), and the optimal values for τ^2 are much less dependent on the sample size than those for the spherical noising. For all three sample sizes we used in our simulations the optimal value for τ^2 was equal to 0.5 (compare to Column 1 in Table 3.3 for spherical noising). For the outer bagged regularized estimator we find that the adaptive target matrix needs to be recomputed on each bootstrap sample, because adding $Cov(F^{(j)}_{n(n)})$ and $\hat{\Sigma}_\epsilon$ computed from the spectral decomposition of the original sample variance-covariance matrix can result in a poorly invertible or even singular matrix. Employing different target matrices each computed based on the spectral decomposition of the corresponding $Cov(F^{(j)}_{n(n)})$ leads to reasonable but still highly variable performance (the variance of the relative squared errors is large, especially for small sample sizes; see Column 2 of Table 3.6). The optimal values for τ^2 are different from those for adaptive augmented bootstrap and depend on the sample size (0.5, 0.6 and 0.8 for sample sizes $10p$, $1.1p$ and $0.5p$ respectively).

Another alternative for the target matrix choice concerns the bagged regularized estimator. Instead of using the target matrix $\hat{\Sigma}_\epsilon$ computed from the spectral decomposition of the sample variance-covariance matrix $Cov(F_n)$ (which leads to a performance indistinguishable from the Adaptive AB), or target matrices computed on each bootstrap resample, as in the outer bagging scheme described above, one can use (prior to inversion) a common target matrix computed relative to the spectral decomposition of the bagged variance-covariance $\frac{1}{m}\sum_{j=1}^{m} Cov(F^{(j)}_{n(n)})$. Using this common target leads to substantial improvements for all sample sizes, especially the smallest (see Column 3 of Table 3.6). We find this observation worth of additional investigation, which we do not pursue here, because it is not directly relevant to the augmented bootstrap methodology.

In addition to modifying the noise variability structure of the AB as proposed by B. Li, another avenue to pursue is to use *a priori* information to tailor such structure to what is known about the sources of error for a given experimental technology. In other words, the AB could be thought of as a tool to simulate

technical replicates of data derived, for instance, from one of the contemporary high-throughput genomic platforms.

3.5 Computational burden

How computationally heavy is AB in comparison to other resampling based approaches, such as bagging? We can derive some indications from the simulations performed in Section 3.2 and the applications in Section 3.3.

For our simulated data with $p = 100$ and $n = 10p = 1000$, the AB method implemented in R with sample size multiplier $m = 70$ (and regardless of the smoothing variance τ^2) takes approximately 6 seconds to produce $\hat{\Sigma}^{-}_{AB}$ (on a Debian lenny/sid workstation, 3.2GHz CPU, 1GB memory). By comparison, bagging the Moore-Penrose inverse of the sample variance-covariance matrix in R to produce $\hat{\Sigma}^{-}_{BB}$ (Schäfer and Strimmer, 2005a,b) on the same simulated data, and using $m = 70$ resamples (so that the overall "amount of resampling" is the same as for AB), takes approximately 2 seconds. This can be explained as follows: implementing the AB method requires (i) resampling mn points from the data; (ii) generating $mn \cdot p$ draws from a univariate $N(0, \tau^2)$ for the augmentation; (iii) computing the inverse of a $p \times p$ variance-covariance matrix derived from mn points. Bagging does not involve (ii), and requires (i) resampling n points from the data, m times; (iii) computing m pseudoinverses of $p \times p$ variance-covariance matrices, each derived from n points, and averaging these matrices. While (i) is equivalent for the two methods, and (iii) much less computationally expensive for AB than for BB, (ii) accounts for the fact that AB is slower than BB on our simulated data.

For the yeast stationary phase microarray data described in Section 3.3.2, where $p = 321$ and $n = 10$, the AB method with $m = 1000$ comprised a number of Gaussian draws in the order of 10^6 (compared to 10^7 for the simulated data). Consequently, step (ii) was less expensive and the running time of AB was only 4 seconds. In contrast, with $m = 1000$ step (iii) was much more expensive for BB than for AB, and BB required about 5 minutes. In other words, while AB introduces the additional burden of augmentation, BB is much more sensitive to the "amount of resampling" than AB.

Whether or not n, p and m are such that AB is slower than BB or vice versa, as

shown in Section 3.2, the gains in performance of the former relative to the latter for a given "amount of resampling" are substantial.

In applications similar to the X chromosome inactivation problem described in Section 3.3.1, non-resampling based methods such as the CLC, which specifically exploit the supervised nature of the analysis to overcome undersampling in a high dimensional predictor space, can perform as well as using AB to estimate the inverse predictor variance-covariance matrix. However, these methods can be computationally expensive too. In particular, CLC requires computing several powers of the sample variance-covariance matrix. With the CLC method implemented in R, producing the discriminating direction $\hat{\beta}_{CLC} = \hat{C}\hat{\nu}$ for the Xp22 dataset required approximately 3 seconds, while producing the discriminating direction $\hat{\beta}_{AB} = \hat{\Sigma}^{-}_{AB}\hat{\nu}$ required approximately 4 – so the CLC was faster, but not substantially so.

In conclusion, the "noising rationale" exploited by the AB approach appears to be remarkably effective for estimating Σ^{-1}, at an expenditure similar or lower than the ones characterizing other resampling based or otherwise computationally intensive methods. In particular, the above evaluation of running times shows that the AB is indeed computationally feasible for the large scale datasets that are typical of genomic applications.

CHAPTER 4

Augmented bootstrap for classification and regression trees

Also in this chapter, we draw upon developments published in Tyekucheva and Chiaromonte (2008b).

4.1 Introduction

In Section 3.4.1 we interpreted the AB as a computational stabilization approach combining data multiplication and regularization. The related notions of augmenting and smoothing are very general, and their applicability is not limited to estimation of inverse covariance matrices. Here, we demonstrate how the AB can be applied to mitigate adverse effects of data shortage, i.e. overfitting, on CART (Classification and Regression Trees, Breiman et al (1984)). Considering for simplicity a binary classification problem, with a sample of n labeled training points in a given p-dimensional predictor space, CART recursively partitions the space on one or another predictor as to maximize the prevalence of one of the classes in

the resulting regions. The splits can be organized in a binary tree; nodes represent nested regions as one moves down, and are each characterized by *class probabilities* (i.e. class frequencies in the region represented by the node).

CART does not rely on inversion of the sample covariance and can, in principle, be applied even when p exceeds n. However, it is prone to overfitting; unless the data is abundant, trees can be "shallow", thus utilizing splits on a small subset of the available predictors, and unstable, with small changes in the data resulting in vastly different tree topologies. In fact, these shallowness and instability are not unlike the limited and unstable span of a high-dimensional covariance matrix computed from a small number of points.

Many methods have been proposed in the literature to mitigate overfitting and improve the performance of CART, including bagging and shrinking. Bagging (Breiman, 1996) involves creating a tree on each bootstrap sample, say $T(j)$, $j = 1, \ldots, m$, but does not produce an "average tree"; what is bagged are the class predictions resulting from each tree – e.g. with a majority vote rule. In symbols, if $c(x, T) \in \{0, 1\}$ indicates the binary label predicted for a vector $x \in \mathbb{R}^p$ using a tree T, the bagged CART will predict

$$c_B(x) = Ind\left(\frac{1}{m}\sum_{j=1}^{m} c(x, T(j)) \geq 0.5\right)$$

where $Ind(\cdot)$ is the indicator function.

As for shrinking, one of the most commonly applied approaches consists of growing a tree, and then walking down its branches (starting from the root) recursively adjusting the class probabilities in each node towards those in the parent node (Hastie and Pregibon, 1990). If ℓ indicates a generic node, $p_\ell \in [0, 1]$ the probability of class $c = 1$ at the node as originally computed by CART, and $\eta(\ell)$ the node's parent, the recursive adjustment can be expressed as

$$\tilde{p}_\ell = (1 - \lambda)p_\ell + \lambda\tilde{p}_{\eta(\ell)} \propto p_\ell + \left(\frac{\lambda}{1 - \lambda}\right)\tilde{p}_{\eta(\ell)} \tag{4.1}$$

Bagging and shrinking can also be combined (e.g. applying a majority vote to the predictions resulting from a bootstrap ensemble of trees, for each of which class probabilities have been recursively shrunk), although the advantages of bagging

after shrinking have been questioned (Pan, 1999).

The AB applied to CART works with a different logic, as both data multiplication and regularization are implemented *at the level of the training data points* X_i, $i = 1, \ldots, n$ *in the predictor space*. The question is therefore how well can AB do relative to bagging and/or shrinking CART, as described above. Relatedly, one could ask whether the AB noising represents the simulated analogue of a regularization formula for the class probabilities, such as (4.1), or perhaps a non-recursive formula in which the nodes' entropy is altered adjusting probabilities towards a common target $\pi \in [0, 1]$

$$\tilde{p}_\ell = (1 - \lambda)p_\ell + \lambda\pi \propto p_\ell + \left(\frac{\lambda}{1 - \lambda}\right)\pi \tag{4.2}$$

(π here could be 0.5, or possibly the probability of class $c = 1$ computed at the root of the tree – i.e. the fraction of "1" labels in the entire training data, prior to any split).

4.2 Simulations

To address these questions, we consider a two-dimensional "checkerboard" simulation. In $\mathbb{R}^2$, we generate $n = 15$ points from a mixture of four Gaussians with common covariance equal to $0.5I$, means $\mu_1 = (1, 1)'$, $\mu_2 = (-1, -1)'$, $\mu_3 = (-1, 1)'$ and $\mu_4 = (1, -1)'$, and weights all equal to 0.25. Points drawn from the first and second components are labeled as class $c = 1$, and points drawn from the third and fourth are labeled as class $c = 0$. While the sample size here is fairly large relative to the dimension, it is still quite small as far as overfitting for CART is concerned. CART, bagged CART, shrunk CART (recursively to the parent), bagged shrunk CART and AB CART are trained on these data, and class predictions are generated for an independent test sample of $k = 200$ draws from the same mixture and labeling system, producing success (correct classification) rates for each of the methods. Table 4.1 contains averages and standard deviations for these success rates over 500 such simulations. The sample size multiplier for AB and the number of bootstrap samples used when bagging are both set to $m = 20$. The smoothing variance τ^2 for the AB and the shrinkage fraction λ used when shrinking are

both optimized on a grid covering the 0–1 range. Interestingly, while bagging and shrinking do not seem to improve the performance of CART on these data, the AB clearly does.

CART	Bag CART	Shr CART	Bag Shr CART	AB CART
0.617 (0.085)	0.617 (0.074)	0.617 (0.085)	0.605 (0.078)	0.724 (0.067)

Table 4.1. Mean (and standard deviation) of the success rates for various tree-based classifiers. Checkerboard simulation data with $p = 2$, $n = 15$ training points, and $k = 200$ test points.

Figure 4.1 shows the AB CART mean success rates as a function of m and τ^2. The behavior here parallels the one observed for the mean relative square errors when estimating inverse covariances (see Figure 3.1); performance increases and levels off as m increases, and improves and then degrades as τ^2 increases.

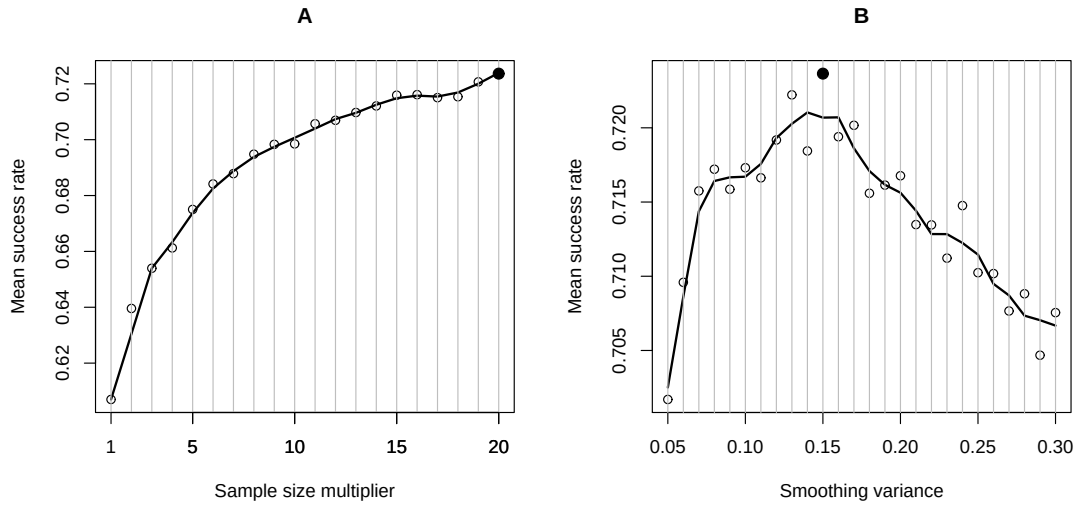

Figure 4.1. AB CART mean success rates for the chekerboard simulated data against the sample size multiplier m (with optimal $\tau^2 = 0.15$; panel A), and the smoothing variance τ^2 (with $m = 20$; panel B). Points represent mean success rates and a lowess smooth is superimposed for visualization. The darkened point shows the maximum success rate achieved with $m = 20$ and $\tau^2 = 0.15$

To investigate whether noising in the predictor space behaves as a simulated analogue of regularization formulae for the class probabilities, such as (4.1) or (4.2), we use again the 2D checkerboard simulation. This time we fix the tree topology (i.e. the splits; we build a tree of depth two with the first order split

on X_1, and the second order splits on X_2). We draw a training sample of size $n = 15$ and compute the class probabilities p_ℓ for each node of the tree. Next, we adjust these class probabilities shrinking recursively to the parent with (4.1), $\tilde{p}_{\ell,PAR}(\lambda)$, and shrinking to a 0.5 target with (4.2), $\tilde{p}_{\ell,0.5}(\lambda)$, using a grid of values for λ – note that for the checkerboard simulation $\pi = 0.5$ corresponds to the "theoretical" class $c = 1$ probability at the root. Finally, for each of a grid of values for τ^2, we noise the given training sample $m = 100$ times. After every such noising, we recompute the class probabilities $\tilde{p}_{\ell,AB}(\tau^2, j)$ for each node of the tree, and we average over repeated noisings to obtain $\tilde{p}_{\ell,AB}(\tau^2) = \frac{1}{m}\sum_{j=1}^{m} \tilde{p}_{\ell,AB}(\tau^2, j)$. Eventually, concentrating on any given node, we have 500 p_ℓ's, 500 $\tilde{p}_{\ell,PAR}(\lambda)$'s for each value of λ, 500 $\tilde{p}_{\ell,0.5}(\lambda)$'s for each value of λ, and 500 $\tilde{p}_{\ell,AB}(\tau^2)$'s for each value of τ^2. Smooth versions of the class probability histograms in a leaf node obtained from 500 simulation rounds are presented in Figure 4.2. The node is the one reached through the splits $X_1 \leq t_1$, followed by $X_2 > t_{1,2}$, where t_1 and $t_{1,2}$ are thresholds computed on each round to optimize separation (this node should contain a prevalence of points belonging to class $c = 1$). One can see how increasing λ or τ^2 modifies the class probabilities, and how different the effects of shrinking formulae and noising in the predictor space are.

In this example we intentionally compared the effects of shrinking formulae and noising with the tree topology fixed. In practical applications, we expect differences to be even more marked, as adjustments with shrinking formulae preserve the topology of the original CART run, while the AB CART can produce a different topology.

4.3 Application to genomic data

In this section we will apply the augmented bootstrap for classification trees to the human-macaque substitution rates data analyzed in Tyekucheva et al (2008).

A better understanding of mutation processes is important for investigating the causes of human genetic diseases and studying the dynamics of molecular evolution. Additionally, identifying and quantifying the effects of genomic features that predict neutral substitution rates is crucial for pursuing a more realistic modeling of neutral vs. selective processes acting on the human genome. Improvements in

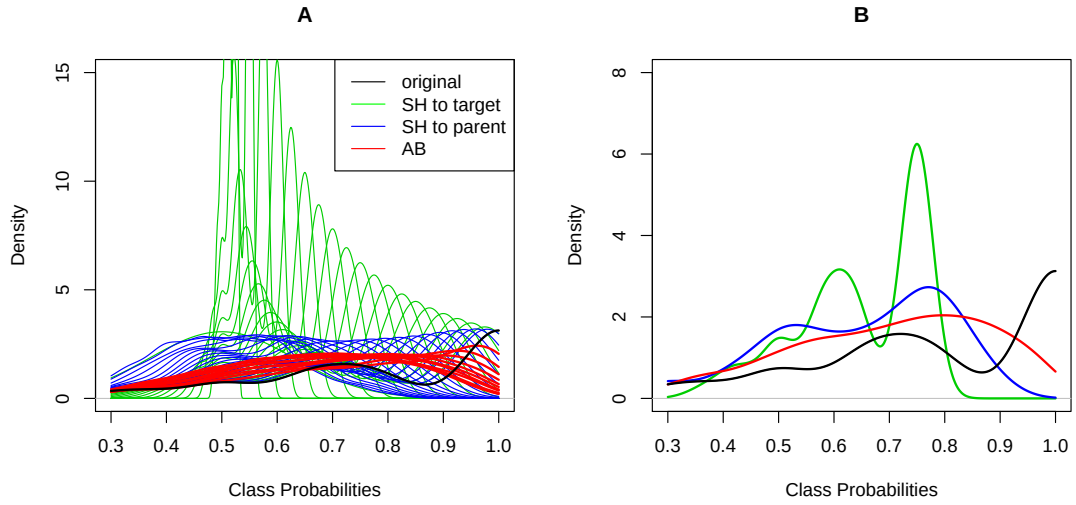

Figure 4.2. Smooth histograms of class probabilities at a leaf node for the chekerboard simulation data (tree topology fixed). Panel A shows probabilities adjusted with shrinking formulae for a range of values of λ (the shrinkage intensity), and those obtained with noising for a range of values of τ^2 (the smoothing variance). Panel B shows representative smooth histograms (shrinkage methods with $\lambda = 0.5$, and noising with $\tau^2 = 0.5$). Black corresponds to the non-modified class probabilities in both panels.

these models may play a role in the development of more accurate computational methods for the identification of functional elements (Taylor et al, 2006).

Rates of nucleotide substitution (divergence) at neutral sites are known to vary within mammalian and other genomes (Wolfe et al, 1989; Lercher et al, 2001; Hardison et al, 2003; Webster et al, 2006). Moreover, such rates have been shown to co-vary with other measures of change in chromosomal DNA, including rates of small insertions and deletions, insertions of transposable elements, and single nucleotide polymorphisms (Hardison et al, 2003; Chiaromonte et al, 2001; Wetterbom et al, 2006; Kvikstad et al, 2007), leading to the hypothesis that some regions in the genome are more prone to evolutionary change of any kind as compared with other regions (Hardison et al, 2003).

Interestingly, neutral substitution rates have also been shown to correlate with GC content, local recombination rates, and distance to telomeres (Hardison et al, 2003; Hellmann et al, 2005). Recombination rate is another important predictor of mammalian divergence, and mechanistically can lead to increased mutation rates through incorrect repair of double-strand breaks (Strathern et al, 1995), although

for humans this has not been demonstrated unequivocally and is still debated (Huang et al, 2005).

While a complete understanding of all biological mechanisms leading to variation in neutral substitution rates across the genome remains elusive, it is plausible that at least some of these mechanisms are conserved over relatively long evolutionary distances. For instance, both mouse-specific and rat-specific substitution rates are positively correlated with rodent-primate substitution rates (Gibbs et al, 2004), suggesting shared mechanisms persisting over ~ 90 million years (Springer et al, 2003). Additionally, a positive correlation exists in substitution rates of homologous X- and Y-chromosomal introns that diverged from each other ~ 100 million years ago (Goetting-Minesky and Makova, 2006).

Relative to previous studies which concentrated on human-mouse (Hardison et al, 2003), mouse-rat (Gaffney and Keightley, 2005) or human-chimpanzee (Hellmann et al, 2005) comparisons, the availability of the macaque genome provides an appealing evolutionary distance to investigate regional variation in the human lineage for the following reasons. First, the human-macaque divergence is smaller than that for human-mouse, and thus can be estimated more accurately. Second, the human-macaque divergence is greater than that for human-chimpanzee, and thus expected to be less affected by biases due to ancestral polymorphism (Li et al, 2002).

In our study (Tyekucheva et al, 2008), we examined regional variation in neutral substitution rates along the human genome utilizing its alignments with the macaque sequence. We used multiple regression techniques to investigate a number of features as predictors of variation in neutral rates, including variables already considered in the literature (e.g., GC content, exon density, SNP density), variables whose definition we modified as to be able to detect subtler associations (e.g., separate male and female recombination rates, distance to telomeres considering positions in both human and macaque), and novel variables (e.g., location on chromosome X vs. autosomes, neutral substitution rates computed from orthologous regions in pair-wise alignments of mouse with rat, and dog with cow). Our regressions explained $\sim 52\%$ of the variation in human-macaque substitution rates and, importantly, they strongly suggested the existence of yet unidentified mutagenic mechanisms whose effects are shared across mammalian genomes. Their effects

are quite substantial compared to the mechanisms captured by the other predictors we considered. Therefore, it is interesting to investigate the genomic regions where neutral substitution rates are relatively high in one species comparison and relatively low in another.

For such analysis we look at the substitution rates from human-macaque and mouse-rat comparisons. We calculated the neutral substitution rates in 1Mb non-overlapping windows (with the human genome as the reference sequence) extracted from 17-way alignments (Blanchette et al, 2004) using the Jukes-Cantor substitution model (Jukes and Cantor, 1969). We define two classes of genomic regions (1Mb windows). The first class comprises regions where human-macaque substitution rates are larger than the 90-th, and mouse-rat rates are smaller than the 10-th quantiles of the corresponding empirical distributions. The second class is defined in the opposite way: human-macaque substitution rates are smaller than the 10-th and mouse-rat rates are larger than the 90-th quantiles. This produces a total of 30 observations (19 in the first class, and 11 in the second). We use six genomic measurements (GC-content, exons density, SNP density, male and female recombination rates, and distance to telomeres) as explanatory variables for our tree-based classifiers. These variables have different scales, so we standardize each of them to have zero mean and unit variance. As compared to our simulations in Section 4.2, undersampling here is even more pronounced (5 observations per predictor vs 7.5 in the simulations). We assess the performance of the traditional, shrunk, bagged, and augmented bootstrap classification trees using leave-one-out cross-validation (results are presented in Table 4.2). Mean and standard deviations of the success rates for bagging and augmented bootstrap are obtained from 50 replications of the corresponding resampling procedures. The augmented bootstrap tuning parameters are optimized on a grid of values (producing $m = 95$, $\tau^2 = 0.5$). The number of bootstrap samples for bagging is chosen to produce the same overall amount of resampling as in the augmented bootstrap with its optimal sample size multiplier. Similarly to our simulation results in Section 4.2, the traditional CART and shrunk CART perform relatively poorly. The tree produced by the traditional CART method is "shallow" and, as a consequence, recursive shrinking to the parent does not improve performance of the classifier. Interestingly, the success rates of the shrunk tree remain the same (0.8) for values of shrinking intensities rang-

CART	Shr CART	Bag CART	AB CART
0.8	0.8	0.87 (0.03)	0.93(0.03)

Table 4.2. Leave-one-out cross-validation success rates for the substitution rates classification problem. For traditional CART and shrinkage with optimal value of the tuning parameter only one success rate is reported. For bagging and augmented bootstrap with optimal values of the tuning parameters mean and standard deviation (parenthetically) of success rates over 50 replications of the resampling procedures are reported.

ing from 0.05 to 0.75 and deteriorate when the intensities are increased further. Unlike in our simulations, bagging does increase success rates, but the augmented bootstrap still shows the best performance. Notably, both the simulations in Section 4.2 and results here suggest that bagging and augmented bootstrap success rates have sizable variability.

Figure 4.3 shows the behavior of the classification success rates as a function of sample size multiplier and smoothing variance. Both plots show patterns similar to the ones obtained from our simulations (see Figure 4.1).

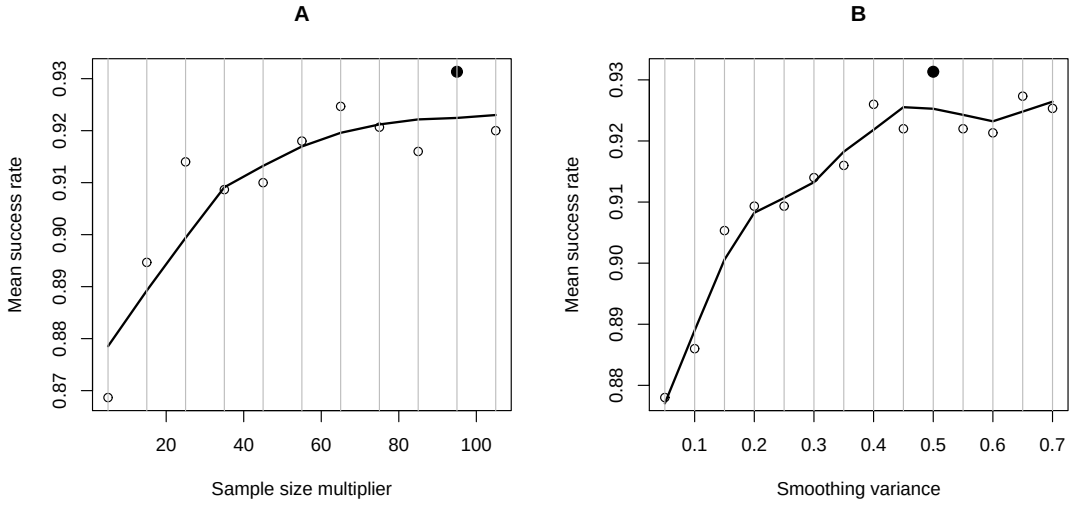

Figure 4.3. AB CART mean success rates for the substitution rates classification problem against the sample size multiplier m (with optimal $\tau^2 = 0.5$; panel A), and the smoothing variance τ^2 (with $m = 95$; panel B). Points represent mean success rates and a lowess smooth is superimposed for visualization. The darkened point shows the maximum success rate achieved with $m = 95$ and $\tau^2 = 0.5$

Although preliminary, our simulations and application to genomic data suggest that the AB, as a computational stabilization approach applicable to different kinds

of statistical procedures, (i) can be as or more effective than existing augmentation and smoothing techniques – taken separately or in combination, and (ii) does not necessarily work as a simulated analogue of these techniques, and may therefore warrant its added computational cost.

CHAPTER 5

Conclusions and future work

In this thesis we proposed a new method - *the augmented bootstrap*. Our method comprises a composition of two well-known regularization approaches; namely, bagging and smoothing. Focusing on a simplified case (functionals expressed as the sums of V-statistics) we derived theoretical conditions under which the augmented bootstrap will not only outperform standard plug-in estimation, but might also outperform bagging and smoothing applied individually. An interesting extension of our theoretical derivations will be characterizing cases in which the augmented bootstrap can reduce the mean squared error (with respect to the plug-in estimator) when bagging and smoothing applied individually have deleterious effects. In such cases MSE reductions arising from the interaction between bagging and smoothing must compensate for MSE increases associated with bagging and smoothing alone. We plan to pursue this line of research.

Using several simulation scenarios and applications to genomic data we found that the augmented bootstrap is an effective and computationally feasible approach. Importantly, the augmented bootstrap can be seen as a computational regularization approach acting at the *level of the data*, which makes it an almost

universal procedure. We can always perturb the data as prescribed by the augmented bootstrap. However, depending on the statistical procedure the augmented bootstrap is overlaid upon, the effects will be different. Noising and data multiplication affect different procedures in different ways. For example, as we have seen in Chapter 3, for estimation of the inverse variance-covariance matrix, the augmented bootstrap is a Monte Carlo approximation to shrinkage (ridge regularization). On the other hand, for the classification trees considered in Chapter 4, the augmented bootstrap does not produce an approximation to shrunk trees. In our analysis in Chapter 4 we considered two types of shrinkage: shrinking to the parent, and shrinking node probabilities towards some fixed (target) probability vector, say the class probabilities observed at the root node. We find it particularly intriguing that not only the augmented bootstrap can help improve performance of various statistical methods, but also allows us to understand how different methods "respond" to perturbations of the data. Further investigations along this lines might lead to a better understanding of which methods are more appropriate given the error structure of the data.

In order to understand and interpret the effects of the augmented bootstrap on CART trees, additional work is needed. A simple example allows us to illuminate some of the open questions. Suppose we restrict our attention to a stump (i.e. a tree with a single split, consisting of a root node and two terminal children nodes), and we fix the variable that is selected for the split. If we use a naïve Bayes classifier as a splitting rule, we can think of the traditional CART method as hard "0-1" thresholding; observations to the right of this threshold will be classified as class 1 and observations to the left will be assigned to class 0 . It can be shown that when the augmented bootstrap is applied, in expectation observations are classified probabilistically, and the "0-1" thresholding is replaced by a Gaussian cumulative distribution function. Of course, in a an actual tree fitting effects are more complex; alteration of a single node/split, almost certainly will affect also its children nodes.

Another example of a statistical procedure that, albeit applicable regardless of the sample size, suffers from overfitting when the data are scarce is support vector machines (SVM). Investigating the behavior of the augmented bootstrap applied to SVM is especially interesting, since they have an intrinsic regularization

component. We plan to pursue this issues in our future work.

Several other important questions remain open. One of them concerns the strategy for choosing the tuning parameters of the augmented bootstrap, i.e. the sample size multiplier and the smoothing variance. In Chapter 3 we compared the augmented bootstrap, where we either optimized performance on a grid of tuning parameters values (see Section 3.2) or fixed those values (see Section 3.3.2), with the shrinkage method implemented in R-package `corpcor`, where the choice of smoothing parameters (i.e. shrinkage intensities) is data-driven. As we noted in Chapter 3, different values for the smoothing parameters might lead to substantial differences in terms of the method's performance despite the fact that the augmented bootstrap is a computational approximation to shrinkage in this case (estimation of the inverse variance-covariance matrix).

These observations suggest that analytically derived data-driven optimal values for tuning parameters might perform poorly, or even become misleading, when the data are undersampled. Interestingly, the data-adaptive choice of target matrix discussed in Section 3.4.2 produced remarkable improvement in the MSE even for small sample sizes. Perhaps data-driven approaches have different degrees of "robustness" to undersampling, and some of them can be used as a compromise between purely data-driven techniques and techniques, that do not allow for any "tuning" based on the observed data.

Whenever possible, we advocate using cross-validation to optimize an appropriate objective function on a grid of values for the tuning parameters. However, cross-validation has its own pitfalls. It can be very computationally intensive, which is not a desirable property. Also, using cross-validation is problematic when there is no obvious choice for the objective function to optimize. In unsupervised problems choosing the objective function might be especially challenging, while it is easier to find a sensible objective function in supervised problems (for example, the observation based mean squared error or classification success rate). However, several objective functions might be explored, for example L_1 and L_2 loss functions. It is well known that the performance of statistical procedures depends on the choice of the loss function. Thus, investigating the behavior of the augmented bootstrap as a function of its tuning parameters under a variety of loss functions is another fruitful avenue to pursue further.

Another important remark is in order: in our discussion of the augmented bootstrap we only considered imposing a Gaussian noise. Notably, our derivations did not rely heavily on the Gaussianity and/or ellipticity of the noise. Similarly to the smoothed bootstrap, the augmented bootstrap noise can be drawn from any appropriate distribution. It is possible that using distributions other than Gaussian to generate noise draws might prove useful in various applications. For example, one might consider drawing the noise from a distribution that models the error structure of the underlying data generating process. Moreover, we considered only continuous random variables. Often predictor vectors will contain some categorical components, say indicator variables. Of course, for categorical random variables imposing noise drawn from the Gaussian distribution, or any other continuous distribution for that matter, is not a sensible choice. For a binary random variable Y, we propose the following noising transformation $Y \to Y^*$:

$$Y^* = |Y - S|,$$

where $S \sim Bernoulli(p_s)$. Then the smoothing variance τ^2 will be a function of the Bernoulli parameter p_s; namely, $\tau^2 = p_s(1 - p_s)$. This noising procedure is equivalent to random label switching with probability p_s, and can be easily generalized to categorical variables with an arbitrary number of label values. Label switching in the discrete/categorical case plays a similar role as convolution with Gaussian noise for the continuous case; it increases the amount of uncertainty in the data.

As far as genomic applications are concerned, with high-throughput sequencing technologies quickly gaining ground (Bennett, 2004; Margulies et al, 2005; Hall, 2007), we expect an abundance of new high-dimensional data to become available. These technologies give rise to many statistical challenges, ranging from *de novo* assembling of sequenced genomes to a whole new perspective on the analysis of the gene expression patterns. We think that applying the augmented bootstrap for analyzing these new data will prove useful in various ways. The notion of the augmented bootstrap as a tool to "simulate" technical replicates of the data might become especially meaningful in these applications.

APPENDIX A

Mean squared error calculations

For the derivations in this appendix we will need the following fact. In full generality, let $\hat{\theta}(\cdot)$ be an estimator for $\theta(\cdot)$. The Mean Square Error (MSE) can always be decomposed as

$$\begin{aligned}
MSE(\hat{\theta}(\cdot)) &= E(\left|\left|\hat{\theta}(\cdot) - \theta(\cdot)\right|\right|^2) \\
&= E(\left|\left|\hat{\theta}(\cdot) \pm E(\hat{\theta}(\cdot)) - \theta(\cdot)\right|\right|^2) \\
&= E(\left|\left|\hat{\theta}(\cdot) - E(\hat{\alpha})\right|\right|^2) + \left|\left|E(\hat{\theta}(\cdot)) - \theta(\cdot)\right|\right|^2 \\
&+ 2(E(\hat{\theta}(\cdot)) - \theta(\cdot))' E(\hat{\theta}(\cdot) - E(\hat{\theta}(\cdot))) \\
&= MSE(\hat{\theta}(\cdot); E(\theta(\cdot))) + \left|\left|E(\hat{\theta}(\cdot)) - \theta(\cdot)\right|\right|^2 \\
&= Var(\hat{\theta}(\cdot)) + Bias^2(\theta(\cdot))
\end{aligned}$$

because $E(\hat{\theta}(\cdot) - E(\hat{\theta}(\cdot))) = 0$.

A.1 Bagging

We will follow notation adopted in Buja and Stuezle (2006), and write:

$$\begin{aligned} A(X) &= A_X \\ B(X,Y) &= B_{XY} \\ B(X,X) &= B_{XX} \\ E(B(X,Y|X)) &= B_X \end{aligned}$$

It is easy to see that the $A(\cdot)$ and $B(\cdot,\cdot)$ terms will be independent only if they do not share any common arguments. Therefore, non-zero covariances will arise only when we consider the terms where some of the arguments are shared:

$$\begin{aligned} Cov(A_X, B_{XY}) &= Cov(A_X, B_X) \\ Cov(B_{XY}, B_{XY'}) &= Var(B_X) \end{aligned}$$

and so on. When the terms do not share common arguments, the covariances will be zero because of independence (we consider an i.i.d. sample).

First we find the variance of $\theta^{(bag)}(F_n)$.

$$Var(\theta^{bag}(F_{n(mn)})) = Var(\theta(F_n)) + g^2 Var(\mathcal{A}_{bag}) + 2gCov(\theta(F_n); \mathcal{A}_{bag})$$

Simple calculations show that:

$$\begin{aligned} Cov(\theta(F_n); \mathcal{A}_{bag}) &= Cov(\frac{1}{n}\sum_i A(X_i) + \frac{1}{n^2}\sum_{i,j} B(X_i, X_j); \frac{1}{n^2}\sum_{i=1}^{n} B(X_i, X_i) \\ &- \frac{1}{n^3}\sum_{i,j=1}^{n} B(X_i, X_j)) \\ &= \frac{1}{n^3}\left[nCov(A_X, B_{XX})\right] \\ &- \frac{1}{n^4}\left[nCov(A_X, B_{XX}) + 2n(n-1)Cov(B_X, B_{XX})\right] \\ &+ \frac{1}{n^4}\left[nVar(B_{XX}) + 2n(n-1)Cov(B_X, B_{XX})\right] \end{aligned}$$

$$
\begin{aligned}
- \quad & \frac{1}{n^5}[4(n^3 - 12n + 8n)Var(B_X) + 2n(n-1)Var(B_{XY}) \\
+ \quad & 4n(n-1)Cov(B_X, B_{XX}) + nVar(B_{XX})] \\
= \quad & \frac{1}{n^2}(Cov(A_X, B_{XX}) - 2Cov(A_X, B_X) + Cov(B_X, B_{XX}) \\
- \quad & 4Var(B_X)) + \mathcal{O}(n^{-3}) \\
= \quad & \frac{1}{n^2}Cov(A_X + 2B_X; B_{XX} - 2B_X) + \mathcal{O}(n^{-3})
\end{aligned}
$$

Similarly, we can show that $Var(\mathcal{A}_{bag}) = \mathcal{O}(n^{-3})$, and consequently $g^2Var(\mathcal{A}_{bag}) = \mathcal{O}(n^{-3})$. Therefore, setting $\alpha_{var}^{(bag)} = \frac{1}{n^2}Cov(A_X + 2B_X; B_{XX} - 2B_X)$ we have:

$$
Var(\theta^{bag}(F_{n(mn)})) \quad = \quad Var(\theta(F_n)) + g\alpha_{var}^{(bag)} + \mathcal{O}(n^{-3})
$$

Next, we find the bias of $\theta^{bag}(F_{n(mn)})$:

$$
\begin{aligned}
Bias(\theta^{bag}(F_{n(mn)})) \quad &= \quad E(\theta(F_n)) + gE(\mathcal{A}_{sm}) - \theta(F) \\
&= \quad Bias(\theta(F_n)) + g(\frac{1}{n}E(B_{XX}) \\
&- \quad \frac{1}{n^3}(nE(B_{XX}) + n(n-1)E(B_{XY}))) \\
&= \quad Bias(\theta(F_n)) + g\left(\frac{1}{n} - \frac{1}{n^2}\right)(E(B_{XX}) - E(B_{XY})) \\
Bias^2(\theta^{bag}(F_{n(mn)})) \quad &= \quad Bias^2(\theta(F_n)) + (g^2 + g)\frac{1}{n^2}\left(E(B_{XX}) + E(B_{XY})\right)^2 + \mathcal{O}(n^{-3}),
\end{aligned}
$$

because $Bias(\theta(F_n)) = \theta(F) + \frac{1}{n}\left(E(B_{XX}) + E(B_{XY})\right)$.

We denote: $\beta_{bias}^{(bag)} = \frac{1}{n^2}\left(E(B_{XX}) + E(B_{XY})\right)$. Therefore, the MSE in Proposition 2.1.1is equal to:

$$
\begin{aligned}
MSE(\theta^{bag}(F_{n(mn)})) \quad &= \quad Var(\theta^{bag}(F_{n(mn)})) + Bias^2(\theta^{bag}(F_{n(mn)})) \\
&= \quad MSE(\theta(F_n)) + g(\beta_{bias}^{(bag)} + \alpha_{var}^{(bag)}) + g^2(\beta_{bias}^{(bag)}) + \mathcal{O}(n^{-3}).
\end{aligned}
$$

A.2 Smoothing

For the derivation of $MSE(\theta^{sm}(F_n))$ we proceed similarly to the calculations made for $MSE(\theta^{bag}(F_{n(mn)}))$. The line of reasoning remains the same; namely, the co-

variances between various terms will be nonzero only if they share same arguments. Additionally to the notation used for the bagged estimator, in an analogous way we denote:

$$\begin{aligned} A_X^{(2)} &= \frac{\partial^2 A}{\partial X^2}(X) \\ A_X^{(4)} &= \frac{\partial^4 A}{\partial X^4}(X) \\ B_{XY}^{(2)} &= \frac{\partial^2 B}{\partial X^2}(X,Y) \\ B_{XX}^{(2)} &= \frac{\partial^2 B}{\partial X^2}(X,X) \\ B_{XY}^{(4)} &= \frac{\partial^4 B}{\partial X^4}(X,Y) \\ B_{XX}^{(4)} &= \frac{\partial^4 B}{\partial X^4}(X,X) \\ B_{XY}^{(2,2)} &= \frac{\partial^4 B}{\partial X^2 \partial Y^2}(X,Y) \end{aligned}$$

The only inconvenience here is that differentiation of $B(\cdot,\cdot)$ eliminates permutation symmetry, unless we consider symmetric mixed derivatives (e.g. $B_{XY}^{(2,2)}$). Lack of symmetry is not explicit from our notation. To keep notation slim and comparable to the notation employed for the bagged estimator, we suggest the following convention:

$$\begin{aligned} B_{X\cdot}^{(2)} &= E\left(\frac{\partial^2 B}{\partial X^2}(X,Y) + \frac{\partial^2 B}{\partial Y^2}(Y,X)|X\right) \\ B_{X\cdot}^{(4)} &= E\left(\frac{\partial^4 B}{\partial X^4}(X,Y) + \frac{\partial^4 B}{\partial Y^4}(Y,X)|X\right) \end{aligned}$$

The derivation of $MSE(\theta^{sm}(F_n(\tau^2));\tau)$ is similar to that of $MSE(\theta^{sm}(F_n(\tau^2));g)$. We derive expressions for the variance and bias squared collecting higher order terms to $\mathcal{O}(\tau^4)$ and $\mathcal{O}(n^{-3})$. We obtain the following expressions:

$$\begin{aligned} Var(\theta^{sm}(F_n(\tau^2))) &= Var(\theta(F_n)) + \tau^4 Var(\mathcal{A}_{sm}) + 2\tau^2 Cov(\theta(F_n);\mathcal{A}_{sm}) \\ &+ 2\tau^4 Cov(\theta(F_n);\mathcal{B}_{sm}) + \mathcal{O}(\tau^5) \\ &= Var(\theta(F_n)) + \tau^4 \beta_{var}^{(sm)} + \tau^2 \alpha_{var}^{(sm)} + \mathcal{O}(\tau^5) + \mathcal{O}(n^{-3}) \\ Bias(\theta^{sm}(F_n(\tau^2))) &= Bias(\theta(F_n)) + \tau^2 E(\mathcal{A}_{sm}) + \tau^4 E(\mathcal{B}_{sm}) + \mathcal{O}(\tau^5) \end{aligned}$$

$$
\begin{aligned}
&= Bias(\theta(F_n)) + \tau^2(\frac{1}{2}E(A_X) + E(B^{(2)}_{XY}) \\
&+ \frac{1}{n}(E(B^{(2)}_{XX}) - E(B^{(2)}_{XY})) \\
&+ \tau^4(\frac{1}{8}E(A^{(4)}_X) + \frac{1}{4}(E(B^{(4)}_{XY}) + E(B^{(2,2)}_{XY})) \\
&+ \frac{1}{n}(E(B^{(4)}_{XX}) + E(B^{(2,2)}_{XX}) - E(B^{(4)}_{XY} - E(B^{(2,2)}_{XY}))) + \mathcal{O}(\tau^5) \\
Bias^2(\theta^{sm}(F_n(\tau^2))) &= Bias^2(\theta(F_n)) + \tau^2\alpha^{(sm)}_{bias} + \tau^4\beta^{(sm)}_{bias} + \mathcal{O}(\tau^5)
\end{aligned}
$$

where:

$$
\begin{aligned}
\alpha^{(sm)}_{var} &= \frac{1}{n}Cov(A_X + 2B_X, A^{(2)}_X + 2B^{(2)}_{X.}) + 2\frac{1}{n^2}(Cov(A^2_X + B^{(2)}_{X.}; B_{XX} - B_X) \\
&+ Cov(A_X + B_X; B^{(2)}_{XX} - B^{(2)}_{X.}) + Cov(B_X; B^{(2)}_{XX}) - 4Cov(B_X; B^{(2)}_{X.}) \\
&+ Cov(B_{XY}; B^{(2)}_{XY})) \\
\beta^{(sm)}_{var} &= \frac{1}{n^4}Var(\sum \frac{\partial^2 B}{\partial X^2_i}(X_i, X_j)) \\
&+ \frac{1}{4}\frac{1}{n}(Cov(A^{(4)}_X + 2B^{(4)}_{X.}; A_X + 2B_X) \\
&+ Cov(B^{(2,2)}_X, 2B_X + 4A_X) + Cov(A^{(2)}_X; A^{(2)}_X + 4B^{(2)}_{X.})) \\
&+ \frac{1}{2}\frac{1}{n^2}(2Cov(A^{(2)}_X; B^{(2)}_X - B^{(2)}_{X.}) + Cov(A_X, B^{(4)}_{XX} - B^{(4)}_{X.} - B^{(2,2)}_{XX}) \\
&+ Cov(B_{XY}; B^{(4)}_{XY} + 2B^{(2,2)}_{XY}) + Cov(A^{(4)}_X; B^4_{XX} - 2B_X) \\
&+ 2Cov(B^{(2,2)}_{XX}; B_{XX}) - Cov(B_X; B^{(2,2)}_{XX} + 4B^{(4)}_{X.})) \\
\alpha^{(sm)}_{bias} &= 2\frac{1}{n}\left(\frac{1}{2}E(A^{(2)}_X) + E(B^{(2)}_{XY}) + \frac{1}{n}\left(E(B_{XX}) - E(B^{(2)}_{XY})\right)\right) \\
&\times (E(B_{XX}) - E(B_{XY})) \\
\beta^{(sm)}_{bias} &= \left(\frac{1}{2}E(A^{(2)}_X) + E(B^{(2)}_{XY}) + \frac{1}{n}\left(E(B_{XX}) - E(B^{(2)}_{XY})\right)\right)^2 \\
&+ 2(\frac{1}{n}(E(B_{XX}) - E(B_{XY}))(\frac{1}{8}E(A^{(4)}_X) + \frac{1}{4}(E(B^4_{XY}) + E(B^{(2,2)}_{XY})) \\
&+ \frac{1}{n}(E(B^{(4)}_{XX}) + E(B^{(2,2)}_{XX}) - E(B^{(4)}_{XY}) - E(B^{(2,2)}_{XY})))
\end{aligned}
$$

The form of the MSE in Proposition 2.9 follows from the above expressions in an obvious way.

A.3 Augmented bootstrap

For the augmented bootstrap we use the notation introduced for bagged and smoothed bootstrap estimators (see Appendices A.1, A.2, and corresponding sections in Chapter 2). Again, we proceed by finding the variance, bias, and bias squared of the estimator:

$$
\begin{aligned}
Var(\theta^{AB}(F_{n(mn)}(\tau^2))) &= Var(\theta(F_n)) + g^2 Var(\mathcal{A}_{bag}) + \tau^4 Var(\mathcal{A}_{sm}) \\
&+ g^2\tau^2 Var(\mathcal{A}_{inter}) + 2Cov(\theta(F_n); g\mathcal{A}_{bag}) \\
&+ 2Cov(\theta(F_n); \tau^2\mathcal{A}_{sm}) + 2Cov(\theta(F_n); \tau^4\mathcal{B}_{sm}) \\
&+ 2Cov(\theta(F_n); g\tau^2\mathcal{A}_{inter}) + 2Cov(\theta(F_n); g\tau^4\mathcal{B}_{inter}) \\
&+ 2Cov(g\mathcal{A}_{bag}; \tau^2\mathcal{A}_{sm}) + 2Cov(g\mathcal{A}_{bag}; \tau^4\mathcal{B}_{sm}) \\
&+ 2Cov(g\mathcal{A}_{bag}; g\tau^2\mathcal{A}_{inter}) + 2Cov(g\mathcal{A}_{bag}; g\tau^4\mathcal{B}_{inter}) \\
&+ 2Cov(\tau^2\mathcal{A}_{sm}; g\tau^2\mathcal{A}_{inter}) + \mathcal{O}(\tau^5) \\
&= Var(\theta(F_n)) + \alpha_{var}^{(bag)} g + \alpha_{var}^{(sm)}\tau^2 + \beta_{var}^{(sm)}\tau^4 \\
&+ 2\frac{1}{n^2}(Cov(A_X + 2B_X; B_{XX}^{(2)} - B_{X\cdot}^{(2)}) \\
&+ Cov(B_{XX} + 2B_X; \frac{1}{2}A_X^{(2)} + B_{X\cdot}^{(2)}) \\
&+ Cov(B_X; B_{X\cdot}^{(2)}))g\tau^2 \\
&+ \frac{1}{n^2}(\frac{1}{2}Cov(A_X + 2B_X, B_{XX}^{(4)} + B_{XX}^{(2,2)} - (B_{X\cdot}^{(4)} + 2B_X^{(2,2)})) \\
&+ Cov(A_X^{(2)} + 2B_{X\cdot}^{(2)}; B_{XX}^{(2)} - B_{X\cdot}^{(2)}))g\tau^4 + \mathcal{O}(\tau^5) + \mathcal{O}(n^{-3}) \\
Bias(\theta^{AB}(F_{n(mn)}(\tau^2))) &= Bias(\theta(F_n)) + gE(\mathcal{A}_{bag}) + \tau^2 E(\mathcal{A}_{sm}) + \tau^4 E(\mathcal{B}_{sm}) \\
&+ g\tau^2 E(\mathcal{A}_{inter}) + g\tau^4 E(\mathcal{B}_{inter}) \\
&= Bias(\theta(F_n)) + g\left(\frac{1}{n} - \frac{1}{n^2}\right)(E(B_{XX}) - E(B_{XY})) \\
&+ \tau^2\left(\frac{1}{2}E(A_X^{(2)}) + E(B_{XY}^{(2)}) + \frac{1}{n}(E(B_{XX}^{(2)}) - E(B_{XY}^{(2)}))\right) \\
&+ \tau^4(\frac{1}{8}E(A_X^{(4)}) + \frac{1}{4}(E(B_{XY}^{(4)}) + E(B_{XY}^{(2,2)})) \\
&+ \frac{1}{n}(E(B_{XX}^{(4)}) + E(B_{XX}^{(2,2)}) - E(B_{XY}^{(4)}) - E(B_{XY}^{(2,2)}))) \\
&+ g\tau^2\left(\frac{1}{n} - \frac{1}{n^2}\right)(E(B_{XX}^{(2)}) - E(B_{XY}^{(2)}))
\end{aligned}
$$

$$
\begin{aligned}
& + \quad g\tau^4 \frac{1}{4}\left(\frac{1}{n}-\frac{1}{n^2}\right)(E(B_{XX}^{(4)})+E(B_{XX}^{(2,2)}) \\
& - \quad E(B_{XY}^{(2)})-E(B_{XY}^{(2,2)})) \\
Bias^2(\theta^{AB}(F_{n(mn)}(\tau^2))) &= Bias^2(\theta(F_n))+(g+g^2)\beta_{bias}^{(bag)}+\tau^2\alpha_{bias}^{(sm)}+\tau^4\beta_{bias}^{(sm)} \\
& + \quad g\tau^2 2(E(B_{XX})-E(B_{XY}))(2\frac{1}{n^2}(E(B_{XX}^{(2)})-E(B_{XY}^{(2)})) \\
& + \quad \left(\frac{1}{n}-\frac{1}{n^2}\right)\left(\frac{1}{2}E(A_X^{(2)})+E(B_{XY}^{(2)})\right)) \\
& + \quad g\tau^4 2(\frac{1}{n^2}((E(B_{XX})-E(B_{XY}))\,(E(B_{XX}^{(4)})+E(B_{XX}^{(2,2)}) \\
& - \quad E(B_{XY}^{(4)})-E(B_{XY}^{(2,2)}))+(E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))^2) \\
& + \quad \left(\frac{1}{n}-\frac{1}{n^2}\right)(E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))\left(\frac{1}{2}E(A_X^{(2)})+E(B_{XY}^{(2)})\right)) \\
& + \quad g^2\tau^2 2\frac{1}{n^2}(E(B_{XX})-E(B_{XY}))(E(B_{XX}^{(2)})-E(B_{XY}^{(2)})) \\
& + \quad g^2\tau^4\frac{1}{n^2}((E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))^2+\frac{1}{2}(E(B_{XX})-E(B_{XY})) \\
& \times \quad (E(B_{XX}^{(4)})+E(B_{XX}^{(2,2)})-E(B_{XY}^{(4)})-E(B_{XY}^{(2,2)})))+\mathcal{O}(n^{-3})
\end{aligned}
$$

We denote:

$$
\begin{aligned}
\alpha_{var}^{(inter)} &= 2\frac{1}{n^2}(Cov(A_X+2B_X;B_{XX}^{(2)}-B_{X\cdot}^{(2)})+Cov(B_{XX}+2B_X;\frac{1}{2}A_X^{(2)}+B_{X\cdot}^{(2)}) \\
&+ \quad Cov(B_X;B_{X\cdot}^{(2)})) \\
\beta_{var}^{(inter)} &= \frac{1}{n^2}(\frac{1}{2}Cov(A_X+2B_X,B_{XX}^{(4)}+B_{XX}^{(2,2)}-(B_{X\cdot}^{(4)}+2B_X^{(2,2)})) \\
&+ \quad Cov(A_X^{(2)}+2B_{X\cdot}^{(2)};B_{XX}^{(2)}-B_{X\cdot}^{(2)})) \\
\alpha_{bias}^{(inter)} &= 2(E(B_{XX})-E(B_{XY}))(2\frac{1}{n^2}(E(B_{XX}^{(2)})-E(B_{XY}^{(2)})) \\
&+ \quad \left(\frac{1}{n}-\frac{1}{n^2}\right)\left(\frac{1}{2}E(A_X^{(2)})+E(B_{XY}^{(2)})\right)) \\
\beta_{bias}^{(inter)} &= 2(\frac{1}{n^2}((E(B_{XX})-E(B_{XY}))\,(E(B_{XX}^{(4)})+E(B_{XX}^{(2,2)})-E(B_{XY}^{(4)}) \\
&- \quad E(B_{XY}^{(2,2)}))+(E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))^2) \\
&+ \quad \left(\frac{1}{n}-\frac{1}{n^2}\right)(E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))\left(\frac{1}{2}E(A_X^{(2)})+E(B_{XY}^{(2)})\right)) \\
\delta_{bias}^{(inter)} &= 2\frac{1}{n^2}(E(B_{XX})-E(B_{XY}))(E(B_{XX}^{(2)})-E(B_{XY}^{(2)}))
\end{aligned}
$$

$$\begin{aligned}\gamma_{bias}^{(inter)} &= \frac{1}{n^2}((E(B_{XX}^{(2)}) - E(B_{XY}^{(2)}))^2 + \frac{1}{2}(E(B_{XX}) - E(B_{XY})) \\ &\times \quad (E(B_{XX}^{(4)}) + E(B_{XX}^{(2,2)}) - E(B_{XY}^{(4)}) - E(B_{XY}^{(2,2)})))\end{aligned}$$

The form of the MSE in Proposition 2.2.1 follows directly from the above expressions.

APPENDIX B

Supplementary tables

n	T	SH	BB	AB
	Σ_1			
$10p$	0.151 (0.004)	0.103 (0.005)	0.241 (0.006)	0.069 (0.001)
$1.1p$	2452.86 (2292.99)	0.238 (0.120))	2.075 (0.268)	0.189 (0.002)
$0.5p$	0.991 (0.108)	0.140 (0.084)	0.724 (0.005)	0.195 (0.002)
	Σ_2			
$10p$	0.148 (0.005)	0.077 (0.004)	0.237 (0.007)	0.069 (0.001)
$1.1p$	2855.96 (3789.26)	0.076 (0.023)	2.054 (0.290)	0.189 (0.003)
$10p$	0.992 (0.095)	0.044 (0.012)	0.728 (0.006)	0.196 (0.003)
	Σ_3			
$10p$	0.152 (0.006)	0.003 (0.0002)	0.242 (0.009)	0.071 (0.002)
$1.1p$	3015.86 (2326.15)	0.004 (0.001)	2.135 (0.267)	0.191 (0.003)
$0.5p$	1.001 (0.082)	0.006 (0.003)	0.721 (0.006)	0.198 (0.002)

Table B.1. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.1$.

n	Σ_1	Σ_2	Σ_3
$10p$	0.2	0.2	0.2
$1.1p$	0.7	0.7	0.8
$0.5p$	0.9	0.9	1

Table B.2. Optimal smoothing variance τ^2 for various choices of Σ, n, $p = 100$ and $\rho = 0.1$.

n	T	SH	BB	AB
	Σ_1			
$10p$	0.151 (0.005)	0.132 (0.004)	0.241 (0.008)	0.089 (0.002)
$1.1p$	3190.21 (2275.27)	2.793 (1.188)	2.122 (0.223)	0.210 (0.003)
$0.5p$	0.984 (0.102)	1.780 (1.909)	0.723 (0.005)	0.194 (0.002)
	Σ_2			
$10p$	0.149 (0.006)	0.121 (0.005)	0.239 (0.009)	0.088 (0.003)
$1.1p$	3176.29 (2452.17)	0.998 (0.317)	2.040 (0.200)	0.209 (0.003)
$10p$	1.005 (0.093)	0.307 (0.130)	0.729 (0.005)	0.190 (0.002)
	Σ_3			
$10p$	0.061 (0.006)	0.241 (0.007)	0.128 (0.013)	0.039 (0.002)
$1.1p$	868.982 (534.185)	0.642 (0.006)	0.962 (0.034)	0.526 (0.008)
$0.5p$	0.989 (0.013)	0.690 (0.007)	0.963 (0.001)	0.654 (0.003)

Table B.3. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of Σ and n. Gaussian simulation data; $p = 100$ and $\rho = 0.9$.

n	Σ_1	Σ_2	Σ_3
$10p$	0.01	0.01	0.01
$1.1p$	0.1	0.1	0.2
$0.5p$	0.1	0.1	0.4

Table B.4. Optimal smoothing variance τ^2 for various choices of Σ, n, $p = 100$ and $\rho = 0.9$.

n	Reg	Bag Reg	AB	Out Bag Reg
Σ_2				
$10p$	0.069 (0.001)	0.069 (0.001)	0.070 (0.001)	0.066 (0.001)
$1.1p$	0.184 (0.002)	0.185 (0.002)	0.187 (0.002)	0.113 (0.001)
$0.5p$	0.183 (0.002)	0.184 (0.002)	0.191 (0.002)	0.096 (0.002)
Σ_3				
$10p$	0.092 (0.001)	0.095 (0.001)	0.091 (0.001)	0.074 (0.001)
$1.1p$	0.215 (0.003)	0.215 (0.003)	0.219 (0.003)	0.152 (0.002)
$0.5p$	0.224 (0.003)	0.223 (0.003)	0.233 (0.003)	0.146 (0.002)
$\tilde{\Sigma}$				
$10p$	0.095 (0.006)	0.095 (0.006)	0.096 (0.006)	0.074 (0.003)
$1.1p$	0.170 (0.013)	0.169 (0.013)	0.173 (0.012)	0.099 (0.007)
$0.5p$	0.155 (0.011)	0.153 (0.011)	0.166 (0.010)	0.083 (0.005)

Table B.5. Mean (and standard deviation) of the Relative Squared Error for various estimators of Σ^{-1} and choices of n; $p = 100$, with $\rho = 0.5$; Σ_2, Σ_3, $\tilde{\Sigma}$ - as described in Section 3.2

APPENDIX C

Selected R code

C.1 Inverse variance-covariance matrix simulations

```
library(MASS)
library(corpcor)
# Parameters.
p= 100

for (n in c(10*p,1.1*p, 0.5*p)) {
  for (rho in c(0.1, 0.5, 0.9)) {
print(c(n,rho))
B = 50
#M = seq(from=10, to =70, by=10)
M=70
SIGMA_X=c(0.01,seq(from=0.1, to=1.2, by=0.1))
K= max(M)
```

```
s_ab_  = matrix(0, length(M), length(SIGMA_X))
s_ab_inv_  = matrix(0, length(M), length(SIGMA_X))
sigma_bag_1 = matrix(0, p,p)
sigma_bag_2 = matrix(0, p,p)

s_hat_inv = list()
s_shrink_inv = list()
s_ab_inv = list()
s_bag1 = list()
s_bag2 = list()

s_hat = list()
s_shrink = list()
s_ab = list()
s_bag = list()

# B data realizations
#Specify model.

#LOW DIM
Sigma= diag(c(1,rep(rho, p-1)))

#GROUPS
Sigma= diag(0,p,p)
Sigma[1:(p/2),1:(p/2)]=rho
Sigma[(p/2+1):p,(p/2+1):p]=rho

qqq = diag((1-rho),p,p)
Sigma = Sigma+qqq

#LOW DIM CORRELATION
Sigma = matrix(rho, p,p)
```

```
qqq = diag((1-rho),p,p)
Sigma = Sigma+qqq

#SPARSE MATRIX.

Sigma= diag(0,p,p)
Sigma[1:10,1:10]=rho
Sigma[11:p,11:p]=diag(rho,(p-10),(p-10))

qqq = diag((1-rho),p,p)
Sigma = Sigma+qqq

Sigma_inv = pseudoinverse(Sigma)
mu = rep(0, p)
for (b in 1:B) {
# Generate sample.

X = mvrnorm(n, mu, Sigma)

# Regular estimators
sigma_hat = var(X)
sigma_hat_inv = pseudoinverse(sigma_hat)
sigma_shrinked = (cov.shrink(X))
sigma_shrinked_inv= pseudoinverse(sigma_shrinked)
#Select bootstrap sample from orig data.

for (i in 1:length(M)) {
   m = M[i]*n
   for (j in 1:length(SIGMA_X)) {

      sigma_x = SIGMA_X[j]
      index<-sample(c(1:n), m, replace=TRUE)
      #Obtain noised sample for aug bootstrap
```

```
      Xstar = X[index, ] + matrix(rnorm(m*p, mean=0,
                          sd = sqrt(sigma_x)), m, p)

      sigma_ab = (var(Xstar))
      sigma_ab_inv = pseudoinverse(sigma_ab)
      s_ab_inv_[i,j]  =  sum((sigma_ab_inv-Sigma_inv)^2)
                              }
                               }
s_ab_inv[[b]] = s_ab_inv_
s_hat_inv[[b]] = sum((sigma_hat_inv-Sigma_inv)^2)
s_shrink_inv[[b]] = sum((sigma_shrinked_inv-Sigma_inv)^2)

## Bagging
  for (k in 1:K) {
      index<-sample(c(1:n), n, replace=TRUE)
      X_bag = X[index,]
      sigma_bag_1 = sigma_bag_1+var(X_bag)
      sigma_bag_2 = sigma_bag_2+pseudoinverse(var(X_bag))

                  }
sigma_bag_1 = pseudoinverse(sigma_bag_1/K)
sigma_bag_2 = sigma_bag_2/K

s_bag1[[b]] = sum((sigma_bag_1-Sigma_inv)^2)
s_bag2[[b]] = sum((sigma_bag_2-Sigma_inv)^2)
                }
}}
```

C.2 CART checkerboard simulations

```
library(MASS)
library(tree)
#Test data
```

```
  n = 200
  n1 = rbinom(1, n, 0.5)
  n11 = rbinom(1, n1, 0.5)
  n12 = n1 - n11
  n2 = n-n1
  n21 = rbinom(1, n2, 0.5)
  n22 = n2 - n21
 Sg = diag(c(0.5,0.5))
 x1<-rbind(mvrnorm(50, mu=c(-1,1), Sigma=Sg),
           mvrnorm(50, mu=c(1,-1), Sigma=Sg))
 x2<-rbind(mvrnorm(50, mu=c(1,1), Sigma=Sg),

 mvrnorm(50, mu=c(-1,-1), Sigma=Sg))
  x_test=rbind(x1,x2)
  test_data = data.frame(X1=x_test[,1], X2=x_test[,2])
    trueclass=as.factor( c(rep(1,100), rep(2,100)))
#Training data.
s_max=500
cart_er= numeric(s_max)
bag_er=numeric(s_max)
bag_shrunk_er=numeric(s_max)
shrink_er = NULL
ab_er=list()
S=0
while ( S!=s_max) {
n = 15
n11=0
n12 =0
n21 =0
n22 = 0

while( min(n11,n12, n21,n22)==0) {
  n1 = rbinom(1,n, 0.5)
```

```
  n11 = rbinom(1, n1, 0.5)
  n12 = n1 - n11
  n2 = n-n1
  n21 = rbinom(1, n2, 0.5)
  n22 = n2 - n21
                                  }
  x1<-rbind(mvrnorm(n11, mu=c(-1,1), Sigma=Sg),
            mvrnorm(n12, mu=c(1,-1), Sigma=Sg))
  x2<-rbind(mvrnorm(n21, mu=c(1,1), Sigma=Sg),
            mvrnorm(n22, mu=c(-1,-1), Sigma=Sg))
  cl=c(rep(1,n1),rep(2,n2))
  x=rbind(x1,x2)
  data=data.frame(cl=as.factor(cl), x)
#ORIGINAL
   u = try(tree(cl~., data, control=tree.control(nrow(data),
    minsize=1, mindev=0)), silent=TRUE)
   if (class(u)=='tree'){
   S=S+1
   print(S)
   pr =predict.tree(u, test_data, type="class")
   cart_er[S] = sum(pr==trueclass)/length(trueclass)
#SHRUNK
  parents=matrix(0, 2^15-1, 2)
  parents[,1]=c(1:(2^15-1))
  v = rep(1:(2^14-1), each=2)
  parents[,2]=c(-1,v)
  err=NULL
  Lambda = seq(0.05,1,by=0.05)
  for (i in c(1:length(Lambda))){
      lambda=Lambda[i]
      frame = unclass(u)$frame
      unclassed_frame = unclass(unclass(u)$frame)
      for (node in attr(unclassed_frame, 'row.names')) {
```

```
          parent_node = parents[as.integer(node),2]
          if (parent_node!=-1) {
             frame[node,6][1:2] =
              lambda*frame[as.character(parent_node),6][1:2] +
               (1-lambda)*frame[node,6][1:2]
                           }
                                                  }
shrunk_tree =unclass(u)
shrunk_tree$frame = frame
class(shrunk_tree)= "tree"
  pr =predict.tree(shrunk_tree, test_data, type="class")
  err =c(err,  sum(pr==trueclass)/length(trueclass))
                          }
  opt_l = which.max(err)
  shrink_er = rbind(shrink_er, err)
#BAGGED
bg=NULL
bg_shrunk = NULL
w = 20
for (i in c(1:w)) {
      index<-sample(c(1:n), n, replace=TRUE)
      bag_data = data[index,]
      bag_tree = tree(cl~. , data=bag_data,
      control=tree.control(nrow(bag_data),minsize=1,
       mindev=0))
      pred = predict.tree(bag_tree, test_data, type='class')
       err_shrunk=NULL
       lambda=Lambda[opt_l]
       frame = unclass(bag_tree)$frame
       unclassed_frame = unclass(unclass(bag_tree)$frame)
       for (node in attr(unclassed_frame, 'row.names')) {
          parent_node = parents[as.integer(node),2]
          if (parent_node!=-1) {
```

```
            frame[node,6][1:2] =
            lambda*frame[as.character(parent_node),6][1:2] +
             (1-lambda)*frame[node,6][1:2]
                              }
                                                          }
       bag_shrunk_tree =unclass(bag_tree)
       bag_shrunk_tree$frame = frame
       class(bag_shrunk_tree)= "tree"
       pred_shrunk = predict.tree(bag_shrunk_tree, test_data,
        type='class')
       bg = cbind(bg, pred)
       bg_shrunk=cbind(bg_shrunk, pred_shrunk)
  }
s1 = function (a) {
     sum(a==1)
                    }
s2 = function (a) {
     sum(a==2)
                    }
one =  apply(bg, 1, s1)
two = apply(bg, 1, s2)
bag_voter = cbind(one, two)
bag_pred_final = apply(bag_voter, 1, which.max)
bag_er[S] = sum(bag_pred_final==trueclass)/length(trueclass)
one =  apply(bg_shrunk, 1, s1)
two = apply(bg_shrunk, 1, s2)
bag_voter = cbind(one, two)
bag_pred_final = apply(bag_voter, 1, which.max)
bag_shrunk_er[S] = sum(bag_pred_final==trueclass)/length(trueclass)

#AB
Tau=seq(0.05,0.3,by=0.01)
err = matrix(-1, 20, length(Tau))
```

```
for (m in c(1:20)) {

multiple_data=NULL
for ( i in c(1:m)){
multiple_data=rbind(multiple_data,data)
}
for (k in c(1:length(Tau))){
  tau= Tau[k]
  er=0
  counter=0
  while( counter!=1){
     noise= mvrnorm(nrow(multiple_data), c(0,0),
     Sigma=matrix(c(tau, 0, 0, tau),2,2))
     data_star = data.frame(cl=multiple_data$cl, X1=
     multiple_data[,2] +noise[,1],
      X2=multiple_data[,3]+noise[,2])

     u_noise = try( tree(cl~., data_star,
      control=tree.control(nrow(data_star),minsize=1,
       mindev=0)), silent=TRUE )
  if (class(u_noise)=='tree') {
  pr =predict.tree(u_noise, test_data, type="class")
  er = er+  sum(pr==trueclass)/length(trueclass)
  counter= counter+1
                                }
                    }
  err[m,k] =er/counter
                      }}
ab_er[[S]] = err
}
}
```

Bibliography

de Angelis D, Young GA (1992) Smoothing the bootstrap. Int Stat Review 60(1):45–56

Bailey JA, Carrel L, , Chakravarti A, Eichler EE (2000) Molecular evidence for a relationship between LINE-1 elements and X chromosome inactivation: The lyon repeat hypothesis. Proc Natl Acad Sci USA 97(12):6634–6639

Bennett S (2004) Solexa ltd. Pharmacogenomics 5(4):433–438

Blanchette M, Kent W, Riemer C, Elnitski L, Smit A, Roskin K, Baertsch R, Rosenbloom K, Clawson H, Green E, Haussler D, Miller W (2004) Aligning multiple genomic sequences with the threaded blockset aligner. Genome Res 14:708–715, DOI 10.1101/gr.1933104

Breiman L (1996) Bagging predictors. Machine Learning 24(2):123–140

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification And Regression Trees. Wadsworth, Belmont, CA

Buhlmann P (2003) Bagging, subagging and bragging for improving some prediction algorithms. In: Recent Advances and Trends in Nonparametric Statistics, Elsevier

Buhlmann P, Yu B (2002) Analyzing bagging. Annals of Statistics 30(4):927–961

Buja A, Stuezle W (2006) Observations on bagging. Statistica Sinica 16:323–351

Carrel L, Park C, Tyekucheva S, Dunn J, Chiaromonte F, Makova K (2006) Genomic environment predicts expression patterns on the human inactive x chromosome. PLoS Genetics 2(9):e151

Cherry JM, Ball C, Weng S, Juvik G, Schmidt R, Adler C, Dunn B, Dwight S, Riles L, Mortimer RK, Botstein D (1997) Genetic and physical maps of saccharomyces cerevisiae. Nature 387:67–73

Chiaromonte F, Yang S, Elnitski L, Yap V, Miller W, Hardison R (2001) Association between divergence and interspersed repeats in mammalian noncoding genomic dna. Proc Natl Acad Sci USA 98:14,503–14,508, DOI 10.1073/pnas.251423898

Chiaromonte F, Cook RD, Li B (2002) Sufficient dimension reduction in regressions with categorical predictors. The Annals of Statistics 30(2):475–497, URL `http://www.jstor.org/stable/2699965`

Cook RD (1998) Regression Graphics: Ideas for Studying Regressions through Graphics. John Wiley and Sons, New York, ISBN 0-471-19365-8

Cook RD, Li B, Chiaromonte F (2007) Dimension reduction in regression without matrix inversion. Biometrika 94(3):569–584

Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press

Efron B (1979) Bootstrap methods:another look at jackknife. Annals of Statistics 7:1–26

Fernholz LT (1983) Von Mises calculus for statistical functionals, Lecture Noes in Statistics, vol 19. Springer-Verlag, New York

Foster M (1961) An application of the Wiener-Kolmogorov smoothing theory to matrix inversion. J SIAM (9):387–392

Freund Y, Schapire R (1996) Experiments with a new boosting algorithm. In Proc 13th International Conference on Machine Learning pp 148–146

Friedman J, Hall P (2000) On bagging and nonlinear estimation. Preprint

Gaffney D, Keightley P (2005) The scale of mutational variation in the murid genome. Genome Res 15:1086–1094, DOI 10.1101/gr.3895005

Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO (2000) Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell 11(12):4241–4257

Gibbs R, Weinstock G, Metzker M, Muzny D, Sodergren E, Scherer S, Scott G, Steffen D, Worley K, Burch P, Okwuonu G, Hines S, Lewis L, DeRamo C, Delgado O, Dugan-Rocha S, Miner G, Morgan M, Hawes A, Gill R, Celera, Holt R, Adams M, Amanatides P, Baden-Tillson H, Barnstead M, Chin S, Evans C, Ferriera S, Fosler C (2004) Genome sequence of the brown norway rat yields insights into mammalian evolution. Nature 428:493–521, DOI 10.1038/nature02426

Goetting-Minesky M, Makova K (2006) Mammalian male mutation bias: impacts of generation time and regional variation in substitution rates. J Mol Evol 63:537–544, DOI 10.1007/s00239-005-0308-8

Gunsalus KC, Ge H, Schetter AJ, Goldberg DS, Han JDJ, Hao T, Berriz GF, Bertin N, Huang J, Chuang LS, Li N, Mani R, Hyman AA, Sonnichsen B, Echeverri CJ, Roth FP, Vidal M, Piano F (2005) Predictive models of molecular machines involved in caenorhabditis elegans early embryogenesis. Nature 436:861–865

Hall N (2007) Advanced sequencing technologies and their wider impact in microbiology. Journal of experimental biology 210(9):1518–1525

Hall P, DiCiccio T, Romano JP (1989) On smoothing and the bootstrap. Annals of Statistics 17(2):692–704

Hardison R, Roskin K, Yang S, Diekhans M, Kent W, Weber R, Elnitski L, Li J, O'Connor M, Kolbe D, Schwartz S, Furey T, Whelan S, Goldman N, Smit A, Miller W, Chiaromonte F, Haussler D (2003) Covariation in frequencies of substitution, deletion, transposition, and recombination during eutherian evolution. Genome Res 13:13–26, DOI 10.1101/gr.844103

Hastie T, Pregibon D (1990) Shrinking trees. AT&T Bell Laboratories, Technical report

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning: Data mining, inference, and prediction New York. Springer-Verlag, New York

Helland IS (1990) On stucture of partial least squares regression. Scandinavian Journal of Statistics 17:97–114

Hellmann I, Prufer K, Ji H, Zody M, Paabo S, Ptak S (2005) Why do human diversity levels vary at a megabase scale? Genome Res 15:1222–1231, DOI 10.1101/gr.3461105

Hoerl AE (1962) Application of ridge analysis to regression problems. Chemical Engineering Progress (58):54–59

Huang S, Friedman R, Yu N, Yu A, Li W (2005) How strong is the mutagenicity of recombination in mammals? Mol Biol Evol 22:426–431, DOI 10.1093/molbev/msi025

Jukes T, Cantor C (1969) Evolution of protein molecules. Mammalian Protein Metabolism pp 21–123

Kamat AR (1981) Incomplete and absolute moments of the multivariate normal distribution with some applications. Biometrika 40(1/2):20–34

Keleş S, Chun H (2008) Comments on: Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):36–39

Knight K, Bassett GW (2002) Second order improvements of sample quantiles using subsamples

Kvikstad E, Tyekucheva S, Chiaromonte F, Makova K (2007) A macaque's-eye view of human insertions and deletions: differences in mechanisms. PLoS Comput Biol 3:1772–1782, DOI 10.1371/journal.pcbi.0030176

Lashkari DA, De Risi JL, McCusker JH, Namath AF, Gentile C, Hwang SY, Brown PO, Davis RW (1997) Yeast microarrays for genome wide parallel genetic and gene expression analysis. PNAS USA 94(24):13,057–13,062

Ledoit O, Wolf M (2003) Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. J Empir Finance 10:603–621

Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannett NM, Harbison CT, Thompson CM, Simon I, Zeitlinger J, Jennings EG, Murray HL, Gordon DB, Ren B, Wyrick JJ, Tagne JB, Volkert TL, Fraenkel E, Gifford DK, Young RA (2002) Transcriptional regulatory networks in saccharomyces cerevisiae. Science 298(5594):799–804

Lercher M, Williams E, Hurst L (2001) Local similarity in evolutionary rates extends over whole chromosomes in human-rodent and mouse-rat comparisons: implications for understanding the mechanistic basis of the male mutation bias. Mol Biol Evol 18:2032–2039

Lezon TR, Banavar JR, Cieplak M, Maritan A, Fedoroff NV (2006) Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. PNAS USA 103:19,033–19,038

Li B (2008) Comments on: Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):19–21

Li KC (1991) Sliced inverse regression for dimension reduction. JASA 86(414):316–327

Li W, Yi S, Makova K (2002) Male-driven evolution. Curr Opin Genet Dev 12:650–656, DOI 10.1016/S0959-437X(02)00354-4

Lyon MF (1998) X-chromosome inactivation: a repeat hypothesis. Cytogenet Cell Genet 80(1-4):133–137

Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes XV, Godwin BC, He W, Helgesen S, Ho CH, Irzyk GP, Jando SC, Alenquer MLI, Jarvie TP, Jirage KB, Kim JB, Knight JR, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, Mcdade KE, Mckenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons JF, Simpson JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothberg

JM (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376 –380

Pan W (1999) Shrinking classification trees for bootstrap aggregation. Pattern Recogn Lett 20(9):961–965

Penrose RA (1955) A generalized inverse for matrices. Proc Cambridge Phil Soc 51:406–413

Quenouille M (1956) Notes on bias estimation. Biometrika 43:353–369

Rao CR, Mitra SK (1971) Generalized Inverse of Matrices and Its Applications. Wiley, New York

Raudys S, Duin RPW (1998) Expected classification error of the Fisher linear classifier with pseudoinverse covariance matrix. Pattern Recogn Lett 19:385–392

Ross MT, Grafham DV, Coffey AJ, Scherer S, McLay K, Muzny D, Platzer M, Howell GR, Burrows C, Bird CP, et al (2005) The DNA sequence of the human X chromosome. Nature 434:325–337

Schäfer J (2008) Comments on: Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):28–30

Schäfer J, Strimmer K (2005a) An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics 21(6):754–764

Schäfer J, Strimmer K (2005b) A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical Applications in Genetics and Molecular Biology 4(1):Article 32

Silverman BW, Young GA (1987) The bootstrap: to smooth or not to smooth? Biometrika 74:469–479

Springer M, Murphy W, Eizirik E, O'Brien S (2003) Placental mammal diversification and the cretaceous-tertiary boundary. Proc Natl Acad Sci USA 100:1056–1061, DOI 10.1073/pnas.0334222100

Strathern J, Shafer B, McGill C (1995) Dna synthesis errors associated with double-strand-break repair. Genetics 140:965–972

Strimmer K (2008) Comments on: Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):25–27

Taylor J, Tyekucheva S, King D, Hardison R, Miller W, Chiaromonte F (2006) Esperr: learning strong and weak signals in genomic sequence alignments to identify functional elements. Genome Res 16:1596–1604, DOI 10.1101/gr.4537706

Tyekucheva S, Chiaromonte F (2008a) Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):1–18

Tyekucheva S, Chiaromonte F (2008b) Rejoinder on: Augmenting the bootstrap to analyze high dimensional genomic data. TEST 17(1):47–55

Tyekucheva S, Makova K, Karro J, Hardison R, Miller W, Chiaromonte F (2008) Human-macaque comparisons illuminate variation in neutral substitution rates. Genome Biology 9(4):R76, DOI 10.1186/gb-2008-9-4-r76, URL `http://genomebiology.com/2008/9/4/R76`

Webster M, Axelsson E, Ellegren H (2006) Strong regional biases in nucleotide substitution in the chicken genome. Mol Biol Evol 23:1203–1216, DOI 10.1093/molbev/msk008

Werner-Washburne M, Braun E, Johnston GC, Singer RA (1993) Stationary phase in the yeast saccharomyces cerevisiae. Microbiol Mol Biol Rev 57(2):383–401

Wetterbom A, Sevov M, Cavelier L, Bergstrom T (2006) Comparative genomic analysis of human and chimpanzee indicates a key role for indels in primate evolution. J Mol Evol 63:682–690, DOI 10.1007/s00239-006-0045-7

Whittaker J (1990) Graphical Models in Applied Multivariate Statistics. Wiley, New York

Wolfe K, Sharp P, Li W (1989) Mutation rates differ among regions of the mammalian genome. Nature 337:283–285, DOI 10.1038/337283a0

Vita

Svitlana Tyekucheva

Education

Ph.D., Statistics, The Pennsylvania State University, University Park, PA USA (2008)

Advisor: Professor Francesca Chiaromonte

M.S., Computer Science, National Technical University of Ukraine "Kyiv Polytechnic Institute", Kiev, Ukraine (2002)

Advisor: Professor Vladimir Podladchikov

B.S., Computer Science, National Technical University of Ukraine "Kyiv Polytechnic Institute", Kiev, Ukraine (2000)

Research Interests

I am interested in developing new and tailoring existing statistical methods to efficiently process and analyze high dimensional genomic data. In particular my areas of interest include applications of dimension reduction, resampling methods, supervised and unsupervised classification, and regression techniques.

Academic Experience

The Pennsylvania State University, University Park, PA USA

Research Assistant **Summer 2003 – present**

- Center for Comparative Genomics and Bioinformatics

Instructor **Spring 2007 – Spring 2008**

- STAT 401. Experimental Methods.

Teaching Assistant **Fall 2003 – Spring 2004**

- STAT 318. Elementary probability.