

# On estimating covariances between many assets with histories of highly variable length

Robert B. Gramacy  
Statistical Laboratory  
University of Cambridge  
bobby@statslab.cam.ac.uk

Joo Hee Lee  
Fidelity Investments  
London  
joohee.lee@uk.fid-intl.com

March 10, 2008

## Abstract

Quantitative portfolio allocation requires the accurate and tractable estimation of covariances between a large number of assets, whose histories can greatly vary in length. Such data are said to follow a monotone missingness pattern, under which the likelihood has a convenient factorization. Upon further assuming that asset returns are multivariate normally distributed, with histories at least as long as the total asset count, maximum likelihood (ML) estimates are easily obtained by performing repeated ordinary least squares (OLS) regressions, one for each asset. Things get more interesting when there are more assets than historical returns. OLS becomes unstable due to rank-deficient design matrices, which is called a “big  $p$  small  $n$ ” problem. We explore remedies that involve making a change of basis, as in principal components or partial least squares regression, or by applying shrinkage methods like ridge regression or the lasso. This enables the estimation of covariances between large sets of assets with histories of essentially arbitrary length, and offers improvements in accuracy and interpretation. Our methods are demonstrated on randomly generated data, and on real financial time series. An accompanying R package called `monomvn` has been made freely available on CRAN.

**Key words:** financial time series, monotone missing data, maximum likelihood, ridge regression, principal component regression, partial least squares, lasso

# 1 Introduction

Missingness in data, and hence the quest if one should eliminate a part of the data or try and estimate characteristics of it, is common in statistical analysis. The missing observation problem varies in style, depending on the type of data. One example is random missingness, which may stem from erroneous data (Dempster et al., 1977). In financial returns data analysis, however, one problem stands out, which we will refer to as monotone missingness. This happens when the assets of interest have different lengths of financial data. There are several possible ways of dealing with this type of incomplete dataset. One way is by utilizing the portion of data available across all of the assets. Another approach involves estimating the missing portion, called *imputation* (e.g., Little and Rubin, 2002). A third approach is the focus of this paper.

Aside from some glitches in data, which will typically give rise to unrealistic spikes or random missingness in data, the monotone style of missingness in time series data can be grouped into two patterns. The first is where the histories of assets differ due to the fact that they have started being publicly traded at different times. The second is where assets close for various reasons, including corporate actions such as M&A (Merger and Acquisition) activities, or liquidation due to bankruptcy. Both are critical problems to address when conducting multivariate analyses. In this paper, we shall focus mainly on the former. The latter, in absence of the former, can be handled similarly. Handling both types of monotone missingness jointly, and other types of approximately monotone missingness, requires the method of data augmentation (Schafer, 1997; Little and Rubin, 2002) and is beyond the scope of this paper.

Data with arbitrary missingness patterns typically require specialized iterative (even stochastic) estimation algorithms that can be slow and cumbersome to implement. However, data which follow a monotone missingness pattern lead to a likelihood which has a convenient factorization. If we further assume that asset returns are multivariate normally distributed (MVN), with histories at least as long as the total asset count, then maximum likelihood (ML) estimators are easily obtained by performing repeated ordinary least squares (OLS) regressions, one for each asset. The method fails when there are more assets than historical returns. In this case the OLS regressions become unstable due to rank-deficient design matrices. This is sometimes called the “big  $p$  small  $n$ ” problem. It has recently received much attention in the statistics community, with ready applications in bioinformatics and genomics, for example. In the context of estimation for data with a monotone missingness pattern, it can severely limit applicability to cases with a small to modest level of missingness.

In financial applications, where there may be more assets than there are historical price observations for (some of) the assets, this essentially means that the method cannot be applied on the full set of assets of interest. This paper explores remedies to this problem. We aim to develop a method that can be applied in settings where some assets have histories which are shorter than the total number of assets, and even when there are more assets than observations. In short, our solution involves replacing OLS with “parsimonious regressions” that either make a change of basis, as in principal components or partial least squares regression, or apply shrinkage, like ridge regression or the lasso. This enables the estimation of covariances between large sets of assets with histories of essentially arbitrary (and uneven) length. Even in situations where OLS would have been sufficient, we find that the more parsimonious approach can offer improvements in accuracy and interpretation.

The remainder of the paper is organized as follows. Section 2 defines the monotone pattern for missing data, derives the corresponding factorized likelihood, and gives an algorithm of repeated regressions to analytically find a ML estimator for the case where the sampling

distribution is assumed to be multivariate normal (MVN). Section 3 outlines methods for dealing with the “big  $p$  small  $n$ ” problem in the context of regression with transformed inputs and shrinkage estimators, highlighting the benefits of increased applicability, accuracy, and interpretability obtained with these methods. Section 4 gives the details of an algorithm—for MVN data under a monotone missingness pattern—that combines the method in Section 2 with the parsimonious regressions in Section 3. We briefly describe an implementation which has been made freely available as an R package called `monomvn`. Section 5 shows the method in action on synthetic data and real financial data with large numbers of assets having histories of highly varying length. Our results are benchmarked against several standard comparators, and are accompanied by comments on interpretation and efficiency. Finally, we conclude with a discussion in Section 6 that focuses on the ramifications of applying this technique in the context of mean–variance portfolio rebalancing, and some limitations inherent in taking a maximum likelihood approach.

## 2 Multivariate normal monotone missing data

Let  $\mathbf{Y}$  be a  $n \times m$  matrix of random observations  $Y_{i,j}$  which may not be completely observed. Denote  $y_{i,j} = \text{NA}$  if the  $i^{\text{th}}$  sample of the  $j^{\text{th}}$  covariate is missing. In other words, if the columns of a sampled  $\mathbf{Y}$ :  $y_{:,1}, \dots, y_{:,m}$ , represent a historical return series of assets indexed by  $j$  and a return for asset  $j$  is not available at time  $i$ , then  $y_{i,j} = \text{NA}$ . Observed  $\mathbf{Y}$  are said to follow a *monotone missingness pattern* [e.g., (Schafer, 1997, Section 6.5.1) or (Little and Rubin, 2002, Section 7.4)] if the columns can be arranged so that  $y_{i,j} \neq \text{NA}$  whenever  $y_{i,j+1} \neq \text{NA}$ . Figure 1 illustrates this property diagrammatically. The row dimension  $n$ , of  $\mathbf{Y}$ , is equal to the number of completely observed samples  $n_1$  of  $\mathbf{y}_1 \equiv y_{:,1}$ , the maximally observed column. Similarly, let  $\mathbf{y}_j \equiv y_{1:n_j,j}$  collect the complete data in the  $j^{\text{th}}$  column of  $\mathbf{Y}$ , so that  $n_j \geq n_{j+1}$ .

The monotone missingness patterns considered in this paper are assumed to be *missing*

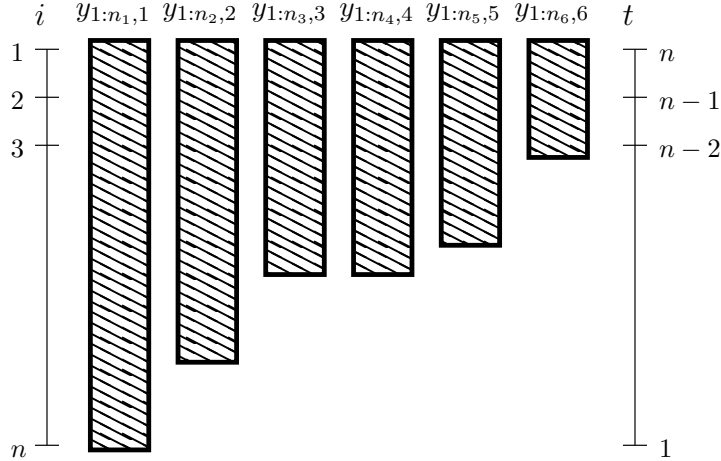


Figure 1: Diagram of a monotone missingness pattern with  $m = 6$  covariates, with a maximum of  $n$  completely observed samples in  $\mathbf{y}_1 = y_{:,1}$ .

*completely at random* (MCAR) in that the pattern of missingness neither depends on the observed nor unobserved responses. Note that there may be columns with identical missingness patterns. In the case of asset return series with observed histories going back different amounts of time, the MCAR assumption may be tenuous, but it is commonly asserted anyway (e.g., Stambaugh, 1996). In our notation, the time index ( $t$ ) for an asset's return history would run counter to  $i$ , the index of the rows of  $\mathbf{Y}$ ; i.e.,  $t = n - i + 1$ , as also illustrated in Figure 1.

For parameters  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)$ , the likelihood  $f(\mathbf{Y}|\boldsymbol{\theta})$  can generally be factorized as follows, when the missing data pattern is monotone:

$$f(\mathbf{Y}|\boldsymbol{\theta}) = f(\mathbf{y}_1|\boldsymbol{\theta}_1)f(\mathbf{y}_2|\mathbf{y}_1, \boldsymbol{\theta}_2)f(\mathbf{y}_3|\mathbf{y}_1, \mathbf{y}_2, \boldsymbol{\theta}_2) \cdots f(\mathbf{y}_m|\mathbf{y}_1, \dots, \mathbf{y}_{m-1}, \boldsymbol{\theta}_m).$$

With the appropriate conditioning, the  $y_{i,j}$  are assumed to be independent and identically distributed (i.i.d.), so that

$$f(\mathbf{y}_j|\mathbf{y}_1, \dots, \mathbf{y}_{j-1}, \boldsymbol{\theta}_j) = \prod_{i=1}^{n_j} f(y_{i,j}|y_{i,1}, \dots, y_{i,j-1}, \boldsymbol{\theta}_j). \quad (1)$$

We are concerned with the case where the  $(y_{i,1}, \dots, y_{i,m})$  follow a multivariate normal distribution (MVN) so that the likelihood in (1) also follows a MVN with constant variance and a mean linear in  $y_{i,1}, \dots, y_{i,j-1}$ . Maximum likelihood estimators (MLEs) of  $\boldsymbol{\theta}_j = (\mu_j, \boldsymbol{\Sigma}_{1:j,j})$ ,  $j = 2, \dots, m$ , are obtained by regression on the complete data:

$$\mathbf{y}_j = \mathbf{Y}_j \boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j, \quad \{\epsilon_{i,j}\}_{i=1}^{n_j} \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma_j^2) \quad (2)$$

where  $\boldsymbol{\beta}_j^\top = (\beta_{0,j}, \beta_{1,j}, \dots, \beta_{(j-1),j})$  and  $\mathbf{Y}_j \equiv \mathbf{Y}_{0:(j-1)}^{(n_j)}$  is the  $n_j \times j$  design matrix

$$\mathbf{Y}_j \equiv \mathbf{Y}_{0:(j-1)}^{(n_j)} = \begin{pmatrix} 1 & y_{1,1} & \cdots & y_{1,(j-1)} \\ 1 & y_{2,1} & \cdots & y_{2,(j-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_{n_j,1} & \cdots & y_{n_j,(j-1)} \end{pmatrix}$$

containing an intercept column, and the first  $n_j$  observations of the first  $j - 1$  columns of  $\mathbf{Y}$ . Figure 2 diagrams the design matrix (without the intercept term) and response vector

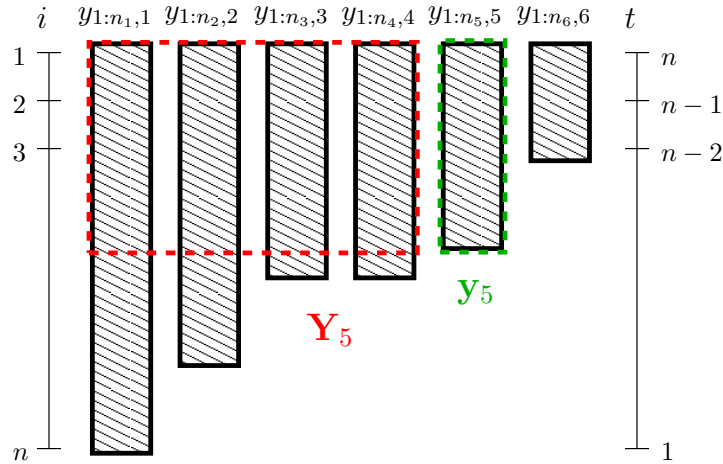


Figure 2: Diagram of the design matrix  $\mathbf{Y}_5$  (without an intercept term) and the response vector  $\mathbf{y}_5$  for the fifth regression involved in maximizing the likelihood of MVN data under a monotone missingness pattern with  $m = 6$  covariates.

involved in one such regression. When  $\text{rank}(\mathbf{Y}_j) = j$ , and particularly when  $n_j > j$ , MLEs are obtainable via the straightforward calculation:

$$\hat{\boldsymbol{\beta}}_j = (\mathbf{Y}_j^\top \mathbf{Y}_j)^{-1} \mathbf{Y}_j^\top \mathbf{y}_j \quad \text{and} \quad \hat{\sigma}_j^2 = \frac{1}{n_j} \|\mathbf{y}_j - \mathbf{Y}_j \hat{\boldsymbol{\beta}}_j\|^2 = \frac{1}{n_j} \sum_{i=1}^{n_j} (y_{i,j} - (\mathbf{y}_i^\top)_{1:n_j} \hat{\boldsymbol{\beta}}_j)^2. \quad (3)$$

Observe that we use a biased estimator for  $\sigma_j^2$ . Then, starting with  $\hat{\boldsymbol{\theta}}_1$  comprising of  $\hat{\mu}_1 = \sum_{i=1}^{n_1} y_{i,1}/n_1$ , and  $\hat{\Sigma}_{1,1} = \sum_{i=1}^{n_1} (y_{i,1} - \hat{\mu}_1)^2/n_1$ , each  $\hat{\boldsymbol{\theta}}_j$  can be estimated conditional on  $\hat{\boldsymbol{\theta}}_{1:(j-1)} = (\hat{\boldsymbol{\mu}}_{1:(j-1)}^\top, \hat{\Sigma}_{1:(j-1),1:(j-1)})$  and estimates of  $\hat{\boldsymbol{\beta}}_j$  and  $\hat{\sigma}_j^2$  as (Stambaugh, 1996):

$$\hat{\mu}_j = \hat{\beta}_{0,j} + \hat{\boldsymbol{\beta}}_{1:(j-1),j}^\top \hat{\boldsymbol{\mu}}_{1:(j-1)} \quad \text{and} \quad \hat{\Sigma}_{1:j,j} = \begin{pmatrix} \hat{\boldsymbol{\beta}}_{1:(j-1),j}^\top \hat{\Sigma}_{1:(j-1),1:(j-1)} \\ \hat{\sigma}_j^2 + \hat{\boldsymbol{\beta}}_{1:(j-1),j}^\top \hat{\Sigma}_{1:(j-1),1:(j-1)} \hat{\boldsymbol{\beta}}_{1:(j-1),j} \end{pmatrix}. \quad (4)$$

When several columns  $\mathbf{y}_\ell$ , say  $\ell = j_1, \dots, j_2$ , have equal lengths of observed histories  $n_\ell$ , it is typical to use a multivariate regression  $(\mathbf{y}_{j_1} \cdots \mathbf{y}_{j_2}) = \mathbf{Y}_{j_1:j_2} \boldsymbol{\beta}_{j_1:j_2} + \boldsymbol{\epsilon}_{j_1:j_2}$  to find  $\hat{\boldsymbol{\beta}}_{j_1:j_2}$  and the empirical variance–covariance matrix  $\hat{\mathbf{V}}_{j_1:j_2,j_1:j_2}$ . Then, several  $\hat{\boldsymbol{\theta}}_{j_1:j_2}$  can be found at once by replacing  $\hat{\boldsymbol{\beta}}_j$  with  $\hat{\boldsymbol{\beta}}_{j_1:j_2}$  and  $\hat{\sigma}_j^2$  with  $\hat{\mathbf{V}}_{j_1:j_2,j_1:j_2}$  in (4). Importantly, if  $\hat{\Sigma}_{1:(j_1-1),1:(j_1-1)}$  and  $\hat{\mathbf{V}}_{j_1:j_2,j_1:j_2}$  are positive definite, then  $\hat{\Sigma}_{1:j_2,1:j_2}$  will be positive definite as well (Stambaugh, 1996).

Calculating such MLEs requires having  $n_j > j$  for all  $j = 1, \dots, m$ . That is, there cannot be an asset whose history is shorter than the number of assets whose histories have greater length. If such were the case, then  $\mathbf{Y}_j$  would not be of full rank, and  $\mathbf{Y}_j^\top \mathbf{Y}_j$  could not be inverted in Eq. (3). This is sometimes referred to in the literature as the problem of regression with “big  $p$  [number of parameters] small  $n$  [number of observations]”. Numerical singularities may arise whenever  $n_j$  is less than, but nearly equal to,  $j$ —especially when  $n$  and  $m$  are large. In the following section we illustrate how these difficulties may be overcome by methods of subset selection, coefficient shrinkage, or the use of principal components.

### 3 Parsimonious regression

In this section, we extract and focus on the subproblem of the linear regression in (2), in terms of a design matrix of  $p$  predictor variables with an intercept term ( $\mathbf{X} \equiv \mathbf{Y}_j$ ) observed for  $n$  cases, with corresponding responses ( $\mathbf{y} \equiv \mathbf{y}_j$ , where  $n \equiv n_j$ ):

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \{\epsilon_i\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2). \quad (5)$$

Ordinary least squares (OLS) gives a MLE of  $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ . Classically, there are two main reasons why one may desire a more parsimonious approach to regression than that provided by OLS. The first is that OLS tends to lead to high variance estimators. The second is a desire for model fits that have high qualitative interpretability, i.e., that describe the data adequately but assume no more causes than will account for the effect. Our reasons for seeking an alternative are related to the former more so than the latter. But, most importantly, we aim to circumvent the problem of having linear dependence in the columns of  $\mathbf{Y}_j$  when  $n_j \leq j$ . In this case, we are faced with an  $n \times p$  design matrix  $\mathbf{X}$  with number of columns  $p$  greater than the number of observations  $n$ , yielding an  $\mathbf{X}^\top \mathbf{X}$  matrix that is singular and cannot be inverted—a so-called “big  $p$  small  $n$ ” ( $p > n$ ) problem. We may even have that  $p \gg n$ , say, when the total number of assets  $m$  is far greater than the number of returns recorded for the asset with the shortest history.

Popular solutions to this problem involve methods of variable selection and coefficient shrinkage. Probably the most straightforward method is *subset selection* (Hastie et al., 2001, Section 3.4.1) which aims to find the model with the “best” size  $k$  (i.e., with  $k \in \{1, \dots, \min(p, n - 1)\}$  covariates). “Best” can be defined in a number of ways, but typically involves  $t$ -tests, or minimizing an estimate of expected prediction error. Searching through all possible subsets quickly becomes infeasible for  $p > 40$ . Larger  $p$  can be handled by greedy methods, but these offer fewer guarantees. Such methods include *forward stepwise selection*

which starts in the null (intercept only) model and sequentially adds predictors, and *backward stepwise selection* which starts at the saturated model (only applicable when  $m < n$ ) and deletes predictors. Hybridizations also exist.

By discarding some predictors, subset selection methods can yield a model which is more interpretable, and may have lower prediction error. But this “discrete” process can produce estimators with high variance. Shrinkage methods are a popular alternative. They are hailed for being more “continuous”, and in some special cases they can have implicit behavior similar to methods like forward selection. The following subsection considers the shrinkage methods of ridge regression, in particular those related to the lasso. In Section 3.2 we consider yet another family of methods which are based on derived input directions: principal components regression, which has connections to ridge regression, and partial least squares regression. These are handy when the predictors are highly correlated.

The parsimonious regression methods outlined in this section have been chosen for familiarity, computational tractability, and implementation. In each case R packages are available on the Comprehensive R Archive Network (CRAN),

<http://cran.R-project.org> (R Development Core Team, 2007),

which provide off-the-shelf implementations that will make for nice subroutines within the framework of constructing estimators for MVN data under monotone missingness. It is typical to first standardize the inputs ( $\mathbf{X}$  and  $\mathbf{y}$ ) as the methods outlined below are not equivariant under re-scaling.

### 3.1 Shrinkage methods: ridge regression, and the lasso

*Ridge regression* and the *lasso* shrink the coefficients of an OLS regression by imposing a

penalty on their size:

$$\hat{\boldsymbol{\beta}}^{(q)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|^q \right\} \quad (6)$$

with  $q = 2$  for ridge regression, and  $q = 1$  for the lasso. The tuning parameter  $\lambda$  controls the amount of shrinkage. Notice that the intercept ( $\beta_0$ ) is left out of the penalty term. Solutions to (6) can be obtained analytically in the case of ridge regression with  $\hat{\boldsymbol{\beta}}^{(2)} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$ . Quadratic programming is required for the lasso. Both methods have interpretations as Bayesian *maximum a posteriori* (MAP) estimators after imposing particular prior distributions. Other choices of  $q > 0$  are also possible, however the constraint region for  $0 < q < 1$  is non-convex, which makes solving the optimization problem more difficult.

For ridge regression, the penalty parameter ( $\lambda$ ) is most advantageously chosen by minimizing cross validation (CV) estimates of predictive error. The commonly used HKB (Hoerl et al., 1975) and L-W (Lawless and Wang, 1976) methods are computationally efficient, but require that  $p < n$  to fit an OLS. The implementation of ridge regression used in this paper comes from the MASS library (Venables and Ripley, 2002) for R in the form of a function called `lm.ridge`.

Though the form of ridge regression and the lasso are similar, there are several important differences. A large  $\lambda$  will cause the ridge estimator  $\hat{\boldsymbol{\beta}}^{(2)}$  to have many coefficients shrunk towards zero. The lasso estimator  $\hat{\boldsymbol{\beta}}^{(1)}$  has a similar effect, but, importantly, may contain many coefficients which are exactly zero—something which is only possible for  $0 < q \leq 1$ . In the Bayesian interpretation, setting  $q \leq 1$  corresponds to choosing a prior which concentrates more mass on small  $|\beta_j|$ , with the most on  $\beta_j = 0$ . In this way, the lasso implements a kind of continuous subset selection. As  $\lambda$  is increased, the  $|\beta_j|$  decrease, eventually increasing the number of them which are identically zero, though this relationship need not be strictly

monotonic.

The implementation of lasso used in this paper is contained in the `lars` package for R (Hastie and Efron, 2007). Efron et al. (2004) show how the lasso, and another method called *forward stagewise*, are special cases of their method of *least angle regression* (LAR). LARS can calculate all possible lasso estimators with computational effort in the same order of magnitude as OLS regression applied to the full set of covariates. CV can be used to select the final model, e.g., using the “one–standard–error” rule (Hastie et al., 2001, Section 7.10), or a more thrifty  $C_p$  (Mallows, 1973) method can be used, but only when  $p < n$ . When applicable, the  $C_p$  method performs nearly as well as CV within the MVN setting with monotone missingness. Madigan and Ridgeway (2004) come to similar conclusions on equally tame benchmarks. However,  $C_p$  has also been criticized for preferring large models (Ishwaran, 2004; Stine, 2004) and for being slightly at odds with LARS (Loubes and Massart, 2004). Since we are mostly interested in applying LARS methods (i.e., lasso) when OLS is not applicable, i.e., when  $p \geq n$ , we shall generally rely on CV to select the final model.

### 3.2 Principal components and partial least squares regression

In situations where there are a large number of highly correlated inputs, a decomposition by principal components (PCs) can be used to select a small number of linear combinations of the original inputs to be used in place of  $\mathbf{X}$ . The related methods of principal component regression (PCR) and partial least squares regression (PLSR) start by performing an orthogonal decomposition of  $\mathbf{X}$ , but differ in how the linear combinations are constructed.

In PCR, *singular value decomposition* (SVD) is performed on  $\mathbf{X}$ , i.e.,  $\mathbf{X} = (\mathbf{UD})\mathbf{V}^\top = \mathbf{TP}^\top$ , where  $\mathbf{U}$  is an  $n \times p$  matrix of left singular vectors describing the “output basis”,  $\mathbf{D}$  is a diagonal matrix containing the corresponding singular values (a square–root of the eigenvalues) in non-decreasing order,  $\mathbf{V}$  is a  $p \times p$  matrix of right singular vectors describing the “input basis”, and  $\mathbf{T}$  and  $\mathbf{P}$  are the so–called *scores* and *loadings* defined by the

decomposition. Next,  $\mathbf{y}$  is regressed on the first  $k$  PCs, i.e., the scores  $\mathbf{T}_{(k)}$ , where the  $(k)$  subscript indicates the extraction of the first  $k$  columns of  $\mathbf{T}$ , i.e., the first  $k$  columns of  $\mathbf{U}$ ,  $\mathbf{V}$ , and the first  $k$  rows/cols of  $\mathbf{D}$ . Since the columns of  $\mathbf{T}$  are orthogonal, the solution is just a sum of univariate regressions. Importantly, the solution can then be written in terms of the coefficients on the predictors in the columns of  $\mathbf{X}$ ,

$$\begin{aligned} \text{(arbitrary scores and loadings)} \quad & \hat{\boldsymbol{\beta}}(k) = \mathbf{P}_{(k)}(\mathbf{T}_{(k)}^\top \mathbf{T}_{(k)})^{-1} \mathbf{T}_{(k)}^\top \mathbf{y} & (7) \\ \text{(from SVD on } \mathbf{X}) \quad & \hat{\boldsymbol{\beta}}^{\text{PCR}}(k) = \mathbf{V}_{(k)} \mathbf{D}_{(k)}^{-1} \mathbf{U}_{(k)}^\top \mathbf{y}, \end{aligned}$$

a vector of length  $p$ . When  $k = p < n$ , the coefficients in (7) are identical to those obtained by OLS. There are many ways of choosing how many components ( $k$ ) to keep in the final model. One way is to consider the relative sizes of the eigenvalues as a proportion of the variation explained by each principal component, and then choose  $k$  so that 80–90% of the variation is explained. A less ad hoc and more reliable—but more computationally intensive—method that can be applied even when  $p \geq n$  involves using CV to estimate predictive error in order to find  $k \in \{1, \dots, \min(p, n - 1)\}$ .

PLSR, by contrast, aims to incorporate information about both  $\mathbf{X}$  and  $\mathbf{y}$  in the scores and loadings—which in this context are often called *latent variables* (LVs)—by proceeding iteratively. The method is initialized with the SVD of  $\mathbf{X}^\top \mathbf{y}$ , thereby including information about the correlation between, and the variance within,  $\mathbf{X}$  and  $\mathbf{y}$ . The scores and loadings obtained by PLSR optimally capture the covariance between  $\mathbf{X}$  and  $\mathbf{y}$ , whereas PCR concentrates only on the variance of  $\mathbf{X}$  (de Jong, 1993). There are several algorithms for obtaining the scores and loadings, but once obtained, the regression coefficients  $\hat{\boldsymbol{\beta}}^{\text{PLSR}}(k)$  in  $\mathbf{X}$ -space are recovered by following (7), and CV can be similarly used to pick  $k$ .

In situations where a minor component of  $\mathbf{X}$  is highly correlated with  $\mathbf{y}$ , PLSR may have a significant advantage over PCR. Otherwise, the methods have a more or less comparable

performance record despite a few operational differences—e.g., PLSR usually needs fewer LVs, but can also yield higher variance estimators of the regression coefficients. Both have behavior similar to other shrinkage methods, particularly ridge regression. For example, it can be shown (Frank and Friedman, 1993) that ridge regression shrinks the coefficients of principal components by a factor of  $d_j^2/(d_j^2 + \lambda)$ , where the  $d_j$  are from the diagonal of  $\mathbf{D}$ , whereas PCR truncates them at  $k$ .

An R package called `pls` (Mevik and Wehrens, 2007) provides a unified implementation of PCR and three algorithms for PLSR (Dayal and MacGregor, 1997; de Jong, 1993; Martens and Næs, 1989), together with built-in facilities for estimating  $k$  via CV.

## 4 The monomvn algorithm

So long as  $n_j > j$  for all  $j = 1 \dots, m$ , and  $n_j \geq n_{j+1}$ , an algorithm for finding the parameters  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  that maximize the MVN likelihood for monotone missing data proceeds as outlined in Section 2. Initialize  $\mu_1$  and  $\Sigma_{11}$  to the sample mean and variance of the first column  $\mathbf{y}_1$  of  $\mathbf{Y}$ , then iterate through the following steps for  $j = 2, \dots, m$ :

1. Find the MLEs (3) of  $\boldsymbol{\beta}_j$  and  $\sigma_j^2$  in a regression (2) of  $\mathbf{y}_j$  onto the first  $j - 1$  columns of  $\mathbf{Y}$  (as predictors), using only the first  $n_j$  observations;
2. Obtain the MLEs of  $\mu_j$  and  $\boldsymbol{\Sigma}_{(1:j),j}$  from  $\hat{\boldsymbol{\mu}}_{1:(j-1)}$ ,  $\hat{\boldsymbol{\Sigma}}_{1:(j-1),1:(j-1)}$ ,  $\hat{\boldsymbol{\beta}}_j$  and  $\hat{\sigma}_j^2$  as in (4).

If any  $n_j \leq j$ , then we have a “big  $p$  small  $n$ ” problem, and the standard regression in step 1 above cannot be performed. In practice, it may be that  $n_j > j$  and still there are columns of the design matrix which are not linearly independent, and so it is not of full rank. The chances that this may happen become increasingly more likely as  $j$  approaches  $n_j$  when finite (double-precision) computer representations make it so that the design matrix is numerically rank deficient. Both issues are addressed simultaneously by instead performing one of the

parsimonious regressions outlined in Section 3. Then step 2 can proceed as usual. Observe that this approach also enables estimation when there are more assets than observations ( $m > n$ ).

Even when parsimonious regression is not strictly necessary, it can aid in interpretation, and possibly even yield more accurate and lower variance estimators. The lasso and the other LARS methods can choose to shrink  $\beta$  so that only the intercept term is nonzero. This enables the detection of zeros in the MVN covariance matrix  $\Sigma$ . In other words, it can be used as a test, of sorts, for independence between assets.

Towards building a more efficient and interpretable estimator, one may consider applying a parsimonious regression for every iteration of step 1 above. Alternatively, one could determine a threshold, say  $p$ , representing a proportion of rows to columns in the design matrix past which a parsimonious regression is applied regardless. I.e., when  $n_j \leq pj$ , for  $0 \leq p \leq 1$ . The  $p = 0$  case corresponds to always using a parsimonious method, and  $p = 1$  reverts to applying one only when necessary. In Section 5 we show how easy it is to establish reliable rules of thumb for choosing  $p$ .

Finally, an R package called `monomvn` (Gramacy, 2007) has been made freely available through CRAN. It implements the algorithm described in this section, and supports all of the parsimonious regression methods outlined in Section 3 via the stand-alone packages outlined therein. Two forms of CV are supported for choosing the number of components in the parsimonious regression: random 10-fold and (deterministic) leave-one-out (LOO). A  $p$  argument facilitates parsimonious regression modeling, as described above.

## 5 Empirical results

In this section, the `monomvn` methods are illustrated and validated on real and synthetic data. Kullback–Leibler (KL) divergence is used as the main metric for comparisons. For arbitrary

distributions with probability distribution functions (PDFs)  $p$  and  $q$ , the KL divergence between  $p$  and  $q$  is defined as

$$D_{\text{KL}}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)}.$$

In the particular case where  $p$  is the estimated MVN with parameters  $\hat{\boldsymbol{\mu}}$  and  $\hat{\boldsymbol{\Sigma}}$  and  $q$  is the “true” parameterization with  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , the KL divergence can be shown to be:

$$D_{\text{KL}}(\text{MVN}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \parallel \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2} \left( \log \frac{|\boldsymbol{\Sigma}|}{|\hat{\boldsymbol{\Sigma}}|} + \text{tr}(\boldsymbol{\Sigma}^{-1} \hat{\boldsymbol{\Sigma}}) + (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}) \right).$$

To investigate how closely an estimator can capture the mean  $\boldsymbol{\mu}$ , and separately the variance  $\boldsymbol{\Sigma}$ , we use the root mean squared error (RMSE):

$$\text{RMSE}(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \sqrt{\frac{1}{m} \sum_{j=1}^m (\hat{\mu}_j - \mu_j)^2}, \quad \text{and} \quad \text{RMSE}(\hat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{m^2} \sum_{i,j=1}^m (\hat{\Sigma}_{i,j} - \Sigma_{i,j})^2}.$$

For the most part, the comparisons to follow focus on highlighting the relative strengths and weaknesses as a function of the choice of parsimonious regression method applied within the `monomvn` algorithm. Additionally, two simpler methods are devised as calibration tools, and to illustrate the advantage of the `monomvn` approach over those which do not leverage the structure of the monotone missingness pattern. The simplest comparator is called “complete”, where  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are estimated using only the portion of data available across all assets, i.e., only the completely observed returns. Put yet another way: only the first  $n_m$  rows of  $\mathbf{Y}$  are used. Another comparator is “observed” which uses all of the available data in an obvious but naïve way:

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} y_{k,j} \quad \text{and} \quad \hat{\Sigma}_{i,j} = \frac{1}{n_j} \sum_{k=1}^{n_j} (y_{k,j} - \hat{\mu}_j)(y_{k,i} - \hat{\mu}_i) \quad \text{for } i = 1, \dots, j. \quad (8)$$

Unfortunately, the covariance matrices provided by the “complete” and “observed” estimators are not guaranteed to be positive–definite (Stambaugh, 1996). Besides meaning that these estimators are invalid, the KL divergence to the true distribution cannot be calculated, and so the RMSE statistics will be our only metric for comparison.

As a final comparator, we consider a method of estimation for incomplete data for arbitrary missingness patterns (Dempster et al., 1977), using the expectation conditional maximization (ECM) algorithm (Meng and Rubin, 1993). Consequently, this method also works when the missingness pattern is monotone, but represents a sort of overkill in this case. Two similar software packages are available for this method when the data is assumed to follow a multivariate normal distribution: the `norm` package (Novo and Schafer, 2002) for R, and `ecmmle` (contained in the `Matlab Financial Toolbox`)<sup>1</sup>. The ECM method iterates until convergence, stopping at a *local* maximum when an improvement threshold is met. As a result, its computational demands and the ultimate optimality of the resulting estimator are sensitive to the initial configuration of the algorithm. Though the missingness pattern may be arbitrary, it is well–known that the method can fail due to convergence issues and/or numerical singularities that can arise due to finite machine representations when more than 15% of the data is missing (see, e.g., the `ecmmle` documentation within `Matlab`). So it cannot handle  $m > n$ , which precludes it from general use in our problem.

## 5.1 Synthetic data

Here, we use a data–generation mechanism provided by the `monomvn` package: `randmvn` generates random samples from a randomly generated MVN distribution with an i.i.d. standard normal mean vector  $\boldsymbol{\mu}$ , and an Inv–Wishart sampled  $\boldsymbol{\Sigma}$ ; `rmono` imposes a uniformly distributed monotone missingness pattern. Table 1 summarizes a comparison between the

---

<sup>1</sup>We prefer `norm` because its core is implemented in compiled Fortran, with an R wrapper. It gives nearly identical results to—but runs more than 20 times faster than—`ecmmle` which is written solely in `Matlab`.

method	KL div		RMSE $\boldsymbol{\mu}$		RMSE $\boldsymbol{\Sigma}$	
	mean	var	mean	var	mean	var
plsr	53.0	2125	0.037	0.00050	0.052	0.0043
pcr	69.0	3249	0.038	0.00054	0.055	0.0049
ridge	45.4	837	0.035	0.00035	0.049	0.0038
lasso	101.9	65966	0.039	0.00043	0.066	0.0075
lar	134.2	125789	0.040	0.00048	0.079	0.0130
fwdstag	104.9	84585	0.039	0.00043	0.066	0.0081
step	258.9	625306	0.041	0.00044	0.096	0.0298
observed			0.067	0.00121	0.099	0.0081
complete			0.289	0.03751	0.302	0.0799

Table 1: Comparison of parsimonious regression ( $p = 1$ ) methods (using 10-fold CV) on randomly generated MVN data ( $n = 1000$  samples,  $m = 100$  dimensions) data with  $\boldsymbol{\mu} \sim N_d(0, 1)$ ,  $\boldsymbol{\Sigma} \sim \text{Inv-Wishart}$  and uniform monotone missingness: means and variances of the KL divergence and RMSE to  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  summarizing 100 repeated trials.

different parsimonious regressions within the `monomvn` algorithm, using randomly generated MVN data with  $m = 100$  and  $n = 1000$ , repeated over 100 trials, each time sampling new  $\boldsymbol{\mu}$ ,  $\boldsymbol{\Sigma}$  and  $\mathbf{Y} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with uniform monotone missingness. Parsimonious regressions were used only when necessary (i.e.,  $p = 1$ ). 10-fold CV was used to choose  $\lambda$  or the number of (principal) components. As can be seen from the table, ridge regression emerges as the clear winner in this comparison, having the lowest average KL divergence and RMSE as well as the lowest variances.

A second experiment was used to determine the optimal setting of the proportion  $p$ . Recall from Section 4 that  $p \in [0, 1]$  determines when a parsimonious method is to be used instead of OLS in the `monomvn` algorithm. The experiment is similar to the previous one, except that  $n$  and  $m$  are varied stochastically with  $m$  uniform in  $\{5, \dots, 100\}$  and  $n|m$  uniform in  $\{\max(10, \lfloor m/2 \rfloor), \dots, md\}$ . Table 2 shows the mean and 90% interval for the optimal  $p$  over 100 repeated trials sampling new  $m$ ,  $n$ , etc., each time. LOO CV was used to choose  $\lambda$ , or the number of (principal) components. The penultimate column in the table shows the proportion of time when  $p = 0$  was better than  $p = 1$ , representing the cases of always using parsimonious regressions, and only when necessary, respectively. For example,

method	optimal $p$			improv	
	5%	mean	95%	$p = 0$	$p = 0.5$
plsr	0.13	0.38	0.62	0.90	0.94
pcr	0.15	0.53	0.84	0.52	0.75
ridge	0.02	0.22	0.62	0.90	0.96
lasso	0.03	0.38	0.76	0.71	0.81
lar	0.06	0.44	0.78	0.63	0.65
stepwise	0.36	0.64	0.91	0.25	0.50

Table 2: Mean and 90% interval for optimal  $p$ , the ratio of columns to rows in the design matrix before switching from OLS to a parsimonious regression. The *improv* columns give the proportion of runs for which  $p = 0$  and  $p = 0.5$  were better than  $p = 1$ , respectively. We repeated this over 100 trials with LOO CV.

PLSR and ridge are regression better with  $p = 0$  ninety percent of the time, while the stepwise and PCR methods are arguably better with  $p = 1$ . The final column in the table illustrates improved performance with a compromise setting of  $p = 0.5$ . All things being equal, a larger  $p$  setting may be preferred for speed reasons.

Due to the limitations of ECM-based methods, like those implemented by `norm` and `ecmmle`, a comparison of `monomvn` to these approaches requires a more controlled experiment. Fixing  $m = 10$  and  $n = 100$ , 1000 repeated experiments similar to the ones described above, with uniform monotone missingness, gave KL divergences with means 3.9 and  $2.4 \times 10^{21}$  with variances 27.0 and  $4.1 \times 10^{45}$  for the `monomvn` (using  $p = 0.5$ ) and `norm` methods, respectively. Medians of 2.2 and 4.8 and a 95% quantile of 12.79 and 32.8 illustrate that while the `monomvn` method is consistently better than the ECM-based `norm` method, the high mean and variance of the `norm` method is due to a few poor, possibly non-converging samples. As  $n$  grows relative to  $m$ , the performance of the methods converge. For example, with  $m = 10$  and  $n = 1000$  the means are 0.50 and 0.62 with variances of 1.8 and 2.6, respectively. However, as the dimensionality ( $m$ ) increases modestly compared to the sample size ( $n$ ), the ECM-based `norm` algorithm consistently diverges. For example, with  $m = 20$  and  $n = 100$  `norm` fails to converge more than 40% of the time.

## 5.2 Real data

From Thomson Financial’s Datastream ([www.datastream.com](http://www.datastream.com)), we have downloaded, in dollar terms, the total returns data of each stock in the Russell 3000<sup>®</sup> Index<sup>2</sup>: 1792 weekly returns between 12/01/1973 and 11/05/2007 for 2894 assets. In order to obtain a set of clean and complete data, each series is tested for illiquidity, completeness, and stationarity, using the following methodology. We removed assets which were marked to market at a frequency other than weekly, to exclude illiquid assets that may exhibit artificial serial correlation (this essentially excludes any stock that has more than two weeks of consecutive unchanging prices at any point in time). Then, an augmented Dicky Fuller test (Dickey and Fuller, 1979) is employed to exclude any of the assets that exhibit non-stationarity (six lags have been tested at the 99% confidence level). A total of 2461 stocks remained after applying these two filtering steps. There are 558 assets with longest history of 1792 returns; the least observed asset has only 76 returns; the overall proportion of missing observations was 0.472.

Our first experiment with this data involves a setup similar to the previous ones for the synthetic data. We further removed any assets with fewer than 635 observations, and then work with only the most recent 635 observations of the assets which remain. This left 617 assets, each with exactly 635 completely observed returns. A full-data mean vector and covariance matrix for these assets can be estimated in the usual way. We then repeatedly interject a uniform monotone missingness pattern and estimate a mean vector and covariance matrix via variants of `monomvn` algorithm and comparators. KL divergence and RMSE are used as metrics to compare the full-data estimates to the estimates under monotone missingness. Table 3 summarizes the results obtained over 30 repeated trials. Parsimonious regressions were only used when necessary (i.e.,  $p = 1$ ). A 10-fold CV was used to choose  $\lambda$  or the number of (principal) components. To summarize the results, ridge regression had

---

<sup>2</sup>The Russel 3000<sup>®</sup> Index represents the broad United States equity universe encompassing approximately 98% of the market.

method	KL div		RMSE $\boldsymbol{\mu}$		RMSE $\boldsymbol{\Sigma}$	
	mean	var	mean	var	mean	var
plsr	1.29e+04	7.04e+06	3.42e-03	4.36e-07	5.65e-04	2.59e-07
pcr	1.37e+04	8.22e+06	3.41e-03	3.11e-07	5.52e-04	2.68e-07
ridge	9.97e+03	4.75e+06	3.41e-03	6.02e-07	5.55e-04	2.51e-07
lasso	1.41e+04	1.29e+07	3.48e-03	1.74e-07	9.43e-04	1.24e-06
lar	1.43e+04	1.31e+07	3.50e-03	1.57e-07	9.45e-04	1.24e-06
step	1.54e+04	1.00e+07	3.50e-03	1.33e-07	7.15e-04	2.17e-07
obs			3.46e-03	1.40e-07	4.22e-04	8.30e-09
complete			1.64e-02	3.12e-06	8.35e-04	2.65e-08

Table 3: Comparison of parsimonious regression ( $p = 1$ ) methods (using 10-fold CV) on real financial data under uniform monotone missingness: means and variances of the KL divergence and RMSE to  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  summarizing 30 repeated trials.

the smallest KL divergence (with smallest variance)—at one standard deviation better than the rest. It also had the smallest mean RMSE for  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , but not the smallest variance. The “observed” and “complete” estimators had a low variance RMSE, but had high mean and never provided a positive definite estimate of  $\boldsymbol{\Sigma}$ .

Making a comparison with `norm` or `ecmmle` similar to the one in Section 5.1 has proved both cumbersome and troublesome; the methods seem unable to handle a dataset of this size with any more than a trivial level of missingness. For example, when the monotone missingness pattern is sampled uniformly, `norm` consistently fails to converge even after thousands of very slow iterations of ECM (each taking several seconds on a 3.2 GHz Xeon).

For our final analysis, we return to the full set of 2461 assets. Here, the “complete” estimator can use only 3% of the observations because the least observed asset has only 76 returns. We consider applying the lasso version of the `monomvn` algorithm to this data, with  $p = 0$ , i.e., always use the lasso (never use OLS). Ridge regression, PCR, and PLSR may be more accurate than the LARS methods (lasso, LAR, forward stagewise, and stepwise) on synthetic benchmarks. However, it can be argued that the LARS methods have better descriptive power because they can provide  $\hat{\boldsymbol{\beta}}$  with many coefficients set to zero. In

the context of the `monomvn` algorithm this means that the MLE  $\hat{\Sigma}$  may have zero entries, indicating marginally uncorrelated assets, and moreover may have block–diagonal structure (or zeros in  $\hat{\Sigma}^{-1}$ ) indicating a pairwise conditional independence of assets. Since ridge regression, PCR, and PLSR always yield  $|\hat{\beta}_i| > 0$ , they would never produce a zero in  $\hat{\Sigma}$ , and so would be less useful for creating such qualitative summaries of the relationships between asset returns. It may be tempting to interject zeros where there are small values in  $\hat{\Sigma}$ , but like the “complete” and “observed” estimators, the resulting matrix would not usually be positive definite. Moreover, classical pairwise tests for independence, say via the Pearson product–moment correlation coefficient, would give unrealistic results. With return histories as short as  $\sim 80$  weeks and estimated correlation less than about 0.2, a simple calculation shows that there would not be enough evidence to reject the hypothesis that the correlation is zero.

The estimator obtained using the lasso on this data yields a  $\hat{\Sigma}$  with 36% of its entries set to zero. Moreover, 50 of its 2641 columns (or 2%) are everywhere zero except in the diagonal position. This means that 36% of asset pairings are marginally uncorrelated. Investigating pairwise correlation between assets, conditional on all of the others, involves looking for zeros in  $\hat{\Sigma}^{-1}$ , of which we find 140 (or 6%). This means that the rows/columns of  $\hat{\Sigma}$  can be reordered so that the matrix has block–diagonal structure, and that the returns of 6% of the assets are conditionally independent. Figure 3 shows histograms summarizing the number of zeros in each column of  $\hat{\Sigma}$  and  $\hat{\Sigma}^{-1}$ . Every column in both matrices had at least one zero entry. The figure clearly illustrates that the resulting correlations can be used to cluster the assets, but this is beyond the scope of this paper.

To wrap up the experiment we downloaded the market returns available from the Russel 3000 index for 1479 (of 1792) contiguous weeks ending 11/5/2007 and used them to create a residual return series for each of the 2461 assets in our study. We then re-ran the lasso experiment, above, to discover that 58% of the asset pairings are marginally uncorrelated and

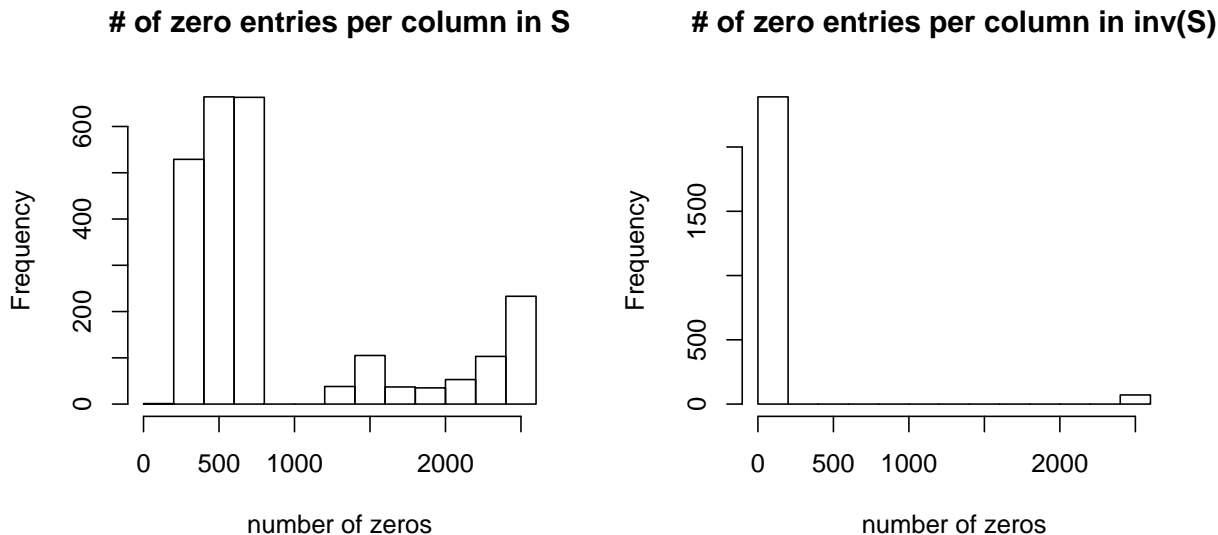


Figure 3: Histograms of the number of zeros in each column of  $\hat{\Sigma}$  (left) and  $\hat{\Sigma}^{-1}$  (right).

14% are conditionally independent when the market is taken into account. The histograms corresponding to this experiment are similar to those for the initial one, in Figure 3, and so they are not reproduced here.

## 6 Discussion

We have shown how the methods of Stambaugh (1996) can be applied for large numbers of assets whose histories are (nearly) unconstrained in length. The key insight is in replacing OLS regressions with more parsimonious ones that either use derived input directions or apply some sort of shrinkage. Whereas Stambaugh demonstrated his methodology on 22 assets, we have shown how the `monomvn` algorithm—essentially the same methodology with a different regression method—can handle thousands. We argued that even when OLS regressions suffice, the more parsimonious ones can offer improvements in both accuracy and interpretation.

Once the maximum likelihood estimators  $\hat{\mu}$  and  $\hat{\Sigma}$  have been obtained, a (Bayesian)

predictive distribution can be constructed, as described by Stambaugh (1996), to project the expectations and covariances one time step into the future. Stambaugh points out that using the predictive expectations and variances in mean–variance portfolio allocation is preferred over using  $\hat{\boldsymbol{\mu}}$  and  $\hat{\boldsymbol{\Sigma}}$  directly, since the latter takes *estimation risk* into account.

But to label this approach as “Bayesian” is an overstatement. The resulting  $\hat{\boldsymbol{\mu}}$  and  $\hat{\boldsymbol{\Sigma}}$  are just point estimates, not posterior distributions. This is indeed a serious limitation of the ML approach. While it is possible to obtain the sampling covariance matrix of  $\hat{\boldsymbol{\mu}}$  analytically, an analytic form for the sampling variability of  $\hat{\boldsymbol{\Sigma}}$  is not known. The bootstrap (e.g. Hastie et al., 2001, Sections 7.11 & 8.2) offers a Monte Carlo method for quantifying the *stability* of  $\hat{\boldsymbol{\Sigma}}$  by producing its component-wise confidence intervals. However, Little and Rubin (2002, Section 7.4.4) make a strong argument in preference for a fully Bayesian approach instead. Facilitating tractable Bayesian estimation for parsimonious regression algorithms, as would be required by `monomvn`, presents a serious challenge. The Bayesian lasso (Park and Casella, 2005) and so-called Bayesian latent factor models (West, 2003), which can be seen as a Bayesian extension of principal components and partial least squares regressions, have received much attention in the recent literature. Exploring the extent to which these can be applied within the `monomvn` algorithm to get samples from the posterior distribution of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  is part of our ongoing work. These samples can be used, in turn, to obtain samples from the posterior predictive distributions to get a truly Bayesian account of the estimation risk in mean–variance portfolio allocation.

Another interesting extension would involve relaxing the assumption of (multivariate) normality, i.e., to decouple the dependence distribution, or *copula* (Sklar, 1957), from the the marginals. There is plenty of evidence in the literature against the assumption of normality for asset returns (e.g. Mills, 1927). Patton (2006) has made promising inroads into applying copulas to a pair of time series under a monotone missingness pattern. Although the theory for copulas (Nelsen, 1999) naturally extends beyond two dimensions, the appli-

cation of the methodology quickly becomes intractable without enforcing severely restrictive assumptions. Our ongoing work includes identifying ways in which the `monomvn` algorithm for high-dimensional estimation under monotone missingness may be extended to support marginal Student  $t$  distributions and various parametric forms of the copula.

## References

- Dayal, B. and MacGregor, J. (1997). “Improved PLS algorithms.” *Journal of Chemometrics*, 11, 1, 73–85.
- de Jong, S. (1993). “SIMPLS: An Alternative Approach to Partial Least Squares Regression.” *Chemometrics and Intelligent Laboratory Systems*, 18, 251–263.
- Dempster, A., Laird, N., and Rubin, D. B. (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society, Series B*, 39, 1, 1–37.
- Dickey, D. and Fuller, W. (1979). “Distribution of the Estimators for Autoregressive Time Series with a Unit Root.” *Journal of the American Statistical Association*, 74, 427–431.
- Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). “Least Angle Regression (with discussion).” *Annals of Statistics*, 32, 2.
- Frank, I. and Friedman, J. (1993). “A Statistical View of Some Chemometrics Regression Tools (with Discussion).” *Technometrics*, 35, 2, 109–148.
- Gramacy, R. B. (2007). *The monomvn Package: Estimation for Multivariate Normal Data with Monotone Missingness*. Statistical Laboratory, University of Cambridge, Cambridge, UK.

- Hastie, T. and Efron, B. (2007). *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 0.9-7.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- Hoerl, A. E., Kennard, R. W., and Baldwin, K. F. (1975). “Ridge Regression: Some Simulations.” *Communications in Statistics*, 4, 105–123.
- Ishwaran, H. (2004). “Discussion of ‘Least Angle Regression’ by B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani.” *Annals of Statistics*, 32, 2, 465–469.
- Lawless, J. and Wang, P. (1976). “A simulation study of ridge and other regression estimators.” *Communications in Statistics – Theory and Methods*, A5, 307–323.
- Little, R. J. and Rubin, D. B. (2002). *Statistical Analysis with Missing Data*. 2nd ed. Wiley.
- Loubes, J.-M. and Massart, P. (2004). “Discussion of ‘Least Angle Regression’ by B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani.” *Annals of Statistics*, 32, 2, 465–469.
- Madigan, D. and Ridgeway, G. (2004). “Discussion of ‘Least Angle Regression’ by B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani.” *Annals of Statistics*, 32, 2, 465–469.
- Mallows, C. (1973). “Some comments on  $C_p$ .” *Technometrics*, 15, 661–675.
- Martens, H. and Næs, T. (1989). *Multivariate Calibration*. Chichester: Wiley.
- Meng, X. and Rubin, D. B. (1993). “Maximum Likelihood Estimation via the ECM algorithm.” *Biometrika*, 80, 2, 267–278.
- Mevik, B.-H. and Wehrens, R. (2007). “The pls Package: Principal Component and Partial Least Squares Regression in R.” *Journal of Statistical Software*, 18, 2.

- Mills, F. (1927). “The Behaviour of Prices.” Tech. rep., National Bureau of Economic Research: New York.
- Nelsen, R. (1999). *An Introduction to Copulas*. New York: Springer–Verlag.
- Novo, A. A. and Schafer, L. (2002). *norm: Analysis of multivariate normal datasets with missing values*. Ported to R by Alvaro A. Novo. Original by Joseph L. Schafer; R package version 1.0-9.
- Park, T. and Casella, G. (2005). “The Bayesian Lasso.” (unpublished), available at <http://www.stat.ufl.edu/~casella/Papers/bayeslasso.pdf>.
- Patton, A. J. (2006). “Estimation of Multivariate Models of Time Series of Possibly Different Lengths.” *Journal of Applied Econometrics*, 21, 147–173.
- Schafer, J. (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall/CRC.
- R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- Sklar, A. (1957). “Fonctions de répartition à  $n$  dimensions et leurs marges.” *Publications de l’Institut Statistique de l’Université de Paris*, 8, 229–231.
- Stambaugh, R. F. (1996). “Analyzing Investments Whose Histories Differ in Length.” *Journal of Financial Economics*, 45, 285–331.
- Stine, R. A. (2004). “Discussion of ‘Least Angle Regression’ by B. Efron, T. Hastie, I. Johnstone, and R. Tibshiran.” *Annals of Statistics*, 32, 2, 465–469.
- Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. 4th ed. New York: Springer. ISBN 0-387-95457-0.

West, M. (2003). “Bayesian factor regression models in the “large p, small n” paradigm.”  
*Bayesian Statistics 7*, 723–732.