Low Complexity Gaussian Latent Factor Models and a Blessing of Dimensionality

Greg Ver Steeg  Aram Galstyan

Abstract

Learning the structure of graphical models from data is a fundamental problem that typically carries a curse of dimensionality. We consider a special class of Gaussian latent factor models where each observed variable depends on at most one of a set of latent variables. We derive information-theoretic lower bounds on the sample complexity for structure recovery that suggest a blessing of dimensionality. With a fixed number of samples, structure recovery for this class using existing methods deteriorates with increasing dimension. We design a new approach to learning Gaussian latent factor models with low computational complexity that empirically benefits from dimensionality. Our approach relies on an information-theoretic constraint to find parsimonious solutions without adding regularizers or sparsity hyper-parameters. Besides improved structure recovery, we also show that we are able to outperform state-of-the-art approaches for covariance estimation on both synthetic data and on under-sampled, high-dimensional stock market data.

1 Introduction

Learning the structure of graphical models, even for Gaussian distributions, is a challenging problem because the space of possible structures grows quickly with the number of variables, \( p \). Common responses to this challenge invoke assumptions of sparsity or low rank. Even so, the sample complexity typically grows with \( p \) making it challenging to apply these methods to high-dimensional but under-sampled data. This type of “big small data” is common in clinical trials of next generation medical instruments or in high-dimensional, non-stationary domains like the stock market where we would like to use as few samples as possible to get an up-to-date model.

In this paper, we introduce a special class of Gaussian latent factor models that have low complexity in three useful senses.

1. **Low sample complexity** We derive an information-theoretic lower bound on sample complexity that suggests a blessing of dimensionality [10]. With a fixed number of latent factors, the number of samples required to accurately recover structure should go down as the number of variables increases. While we are not able to show a matching upper bound on sample complexity, we provide empirical evidence that our approach reflects this blessing of dimensionality and is the only method to do so.

2. **Low computational complexity** We introduce a non-convex learning objective with a quasi-Newton optimization scheme exhibiting a super-linear rate of convergence and an overall time complexity that is linear in the number of variables. Even simple matrix factorization methods are at least quadratic in the number of variables.

3. **Low human complexity** Non-overlapping latent factor models are easy to interpret and popular for exploratory analysis in social science and biology [5]. Our approach uses a novel information-theoretic constraint to encourage non-overlapping factor structure without hyper-parameters or explicit sparsity regularizers.
Our experiments on both synthetic and real-world data demonstrate that the proposed method offers superior performance compared to existing state-of-the-art techniques for both structure recovery and covariance estimation.

2 The sample complexity of Gaussian latent factor models

Capital $X_1$ denotes a continuous random variable whose instances are denoted in lowercase, $x_i$. We abbreviate multivariate random variables, $X \equiv X_1:p \equiv X_1, \ldots, X_p$, with an associated probability density function, $p_X(X_1 = x_1, \ldots, X_p = x_p)$, which is typically abbreviated to $p(x)$, with vectors in bold. Similarly, let $Z$ denote a latent multivariate random variable in $\mathbb{R}^m$. This paper is concerned solely with the case where $X, Z$ define a joint Gaussian distribution. For simplicity we assume all variables have zero mean. Therefore, the covariance matrix for $X$ has components, $\Sigma_{i,j} = \langle X_i X_j \rangle$, where brackets are used for expectation values. Capital letters besides $X$ and $Z$ denote matrices rather than random variables.

How many samples does it take to reliably estimate the structure of a Gaussian graphical model? If we marginalize out the latent factors in a Gaussian model, we still have a distribution over observed variables that is Gaussian and this fully observed setting has been well studied. In general, the number of samples required to estimate the structure of a Gaussian graphical model goes like $d \log p$, where $d$ is the maximum degree of nodes in the graphical model. A method achieving this scaling up to a proportionality constant was recently introduced. Other latent factor modeling approaches also have sample complexity growing with $\log p$. To beat the curse of dimensionality, we introduce a special class of Gaussian latent factor models with lower complexity.

![Diagram](attachment:image.png)

Figure 1: (a) A generic latent factor generative model along with an equivalent characterization. (b) Gaussian latent factor models with non-overlapping structure admit an equivalent information-theoretic characterization.

**Definition Non-overlapping Gaussian latent factor (NGLF) models (Fig. 1(b))** We define a $(p, m)$ NGLF model as a jointly Gaussian distribution with $p$ observed variables, $X_1:p$, and $m$ independent latent variables, $Z_1:m$. Additionally, each $X_i$ has exactly one parent $Z_j$, as in Fig. 1(b).

What if all the relationships among the observed variables are actually due to a fixed number of latent factors? This can considerably simplify the structure learning problem, especially when the number of variables and the maximum degree in the marginal graph become large. We now derive a lower bound on sample complexity for learning the structure of NGLF models that suggests that increasing the dimensionality can actually make learning easier. We follow the construction of information-theoretic sample complexity bounds in. First, we define an ensemble of graphical structures to distinguish among, then we use Fano’s inequality to lower bound the number of samples required to distinguish them with fixed probability of error.

**Theorem 2.1.** For a NGLF model with $p$ variables and $m$ latent factors with $p/m$ children each and AWGN channel from parent to child with signal to noise ratio SNR, the number of samples required to recover the structure of the graphical model with error probability $\epsilon$ obeys the following lower bound.

$$n \geq \frac{2 \left(1 - \epsilon\right) \log \left(\left(\frac{p}{m^{p/m}}\right) \frac{1}{m^{(m-1)/m}}\right) - 1}{(p-1) \log(1 + \text{SNR}^{1/m}) - (m-1) \log(1 + \text{SNR}^{1/m})}$$

(1)

The proof is in Sec. A. The expression in Eq. 1 is plotted for some values in Fig. 2. For a fixed number of latent factors, we see a “blessing of dimensionality” for which the sample complexity goes down...
with increasing dimension, $p$. Intuitively the problem could get easier because more variables provide more signal to reconstruct the fixed number of latent factors. While it is tempting to retrospectively see this as obvious, the same argument could be (mistakenly) applied to other low rank/latent factor models models [6 2 7] that do not enjoy a blessing of dimensionality. We leave the question of whether a blessing of dimensionality is actually achievable in this case to subsequent sections.

A Bayesian network defines a factorization of the joint probability distribution [24]. For example, the expression in Eq. 1 is not very intuitive because it involves the log of a multinomial coefficient. Asymptotics

\[ \log \left( p \right) \approx p \log m + 1/2 \log (p/m) - m/2 \log (m p 2\pi e^2). \]

In the limit of large $p$, we can see from Fig. 2 that the sample complexity lower bound becomes constant. Using Stirling’s approximation, we can derive this asymptotic bound.

\[ n \geq \frac{2(1 - e) \log m}{\log (1 + SNR(1 - 1/m))} \quad (2) \]

Sample complexity upper bound Any method that can recover structure with a fixed number of samples can provide an upper bound. Loose upper bounds can be deduced from more general Gaussian graphical model reconstruction schemes that suggest sample complexity growing like $\log p$. We derive a new, fast approach for recovering NGLF models and show empirically that it exhibits a blessing of dimensionality. I.e., for a fixed number of samples structure recovery improves with $p$, while other methods show no such improvement.

3 Latent factor models via information-theoretic constraints

A Bayesian network defines a factorization of the joint probability distribution [24]. For example, the network in Fig. 1(a) admits the factorization, $p(x, z) = \prod_{i=1}^{p} p(x_i|z) \prod_{j=1}^{m} p(z_j)$. Some types of factorizations can be expressed succinctly in terms of a single information-theoretic functional of the probability density. Multivariate mutual information, historically called total correlation [31], is defined as $TC(Z) = D_{KL}(p(z) || \prod_j p(z_j))$. $TC(Z) = 0$ if and only if the variables are independent. Searching for a representation where $z = Wx$ and $TC(Z) = 0$ is known as (linear) independent component analysis (ICA) [15].

Conditional total correlation is defined as the Kullback-Leibler divergence between the joint distribution and the conditionally independent distribution.

\[ TC(X|Z) = D_{KL} \left( p(x|z) \bigg|\bigg| \prod_{i=1}^{p} p(x_i|z) \right) \quad (3) \]

This quantity is non-negative and zero if and only if all the $X_i$’s are independent conditioned on $Z$. If $Z$ were the hidden source of all dependence in $X$, then $TC(X|Z) = 0$. This equality further implies that $p(x|z) = \prod_i p(x_i|z)$. The Bayesian network depicted in Fig. 1(a) has the property that $TC(X|Z) = 0$ and $TC(Z) = 0$, for example. In fact, we can provide a useful equivalent characterization of this latent factor model in terms of a single information-theoretic functional.
Theorem 3.1. Fig. 1(a) equivalence. The random variables, $X$, $Z$, are described by a directed graphical model where the parents of $X$ are in $Z$ and the $Z$’s are independent if and only if $TC(X|Z) + TC(Z) = 0$.

The proof is straightforward and included in Sec. A.2. If we were to calculate $Z$ as a function of $X$, then by minimizing $TC(Z)$ we would get ICA, but by minimizing $TC(X|Z) + TC(Z)$ we are going further and trying to build a generative factor model for $X$, with success achieved at the global minimum of zero.

Theorem 3.2. Fig. 1(b) equivalence. An NGLF model is equivalent to a jointly Gaussian distribution over $X$, $Z$, where $TC(X|Z) + TC(Z) = 0$ and $\forall i$, $TC(Z|X_i) = 0$.

The proof is included in Sec. A.3. This theorem gives us a condition that moves us from a generic latent factor model in Thm. 3.1 to a structured one. However, unlike typical sparse regularizers, this theorem picks out a special sparsity pattern where each variable has at most one parent. Furthermore, there is no sign of the standard $L_p$ norm or associated hyper-parameters, instead we achieve sparsity through a nonlinear, information-theoretic constraint.

4 Learning the latent factor model

Objective. Typically, to learn a generative model like the one in Fig. 1(a), we would assume it from the outset and then try to maximum the likelihood of the data under this model using an EM procedure. We do not assume a generative model. Instead, we let $Z_j$ be arbitrary functions of the input data and then optimize them to look as close to a generative model as possible using Thm. 3.1.

$$\min_{p(z_j|x)} TC(X|Z) + TC(Z)$$

(4)

Even though $Z$’s are functions of $X$, if we achieve the global minimum of zero, then Thm. 3.1 tells us that we can interpret the resulting distribution as a generative model of the form in Fig. 1(a).

We sketch the main steps in deriving the learning algorithm here, with details provided in Appendix B. First, the objective in Eq. 4 is proportional to $\sum_{i=1}^{p} H(X_i|Z) + \sum_{m=1}^{m} H(Z_i)$. We assume throughout that $X$, $Z$ are jointly Gaussian and the data are standardized so that $\langle X_i \rangle = 0$, $\langle X_i^2 \rangle = 1$.

$$TC(X|Z) + TC(Z) \propto \sum_{i=1}^{p} \frac{1}{2} \log \langle (X_i - \mu_{X|Z})^2 \rangle + \sum_{j=1}^{m} \frac{1}{2} \log \langle Z_j^2 \rangle$$

(5)

The mean of $X_i$ conditioned on $Z$, $\mu_{X|Z}$, is easy to express for Gaussians but hard to calculate because it involves a matrix inversion. We still have not used the constraint that $\forall i$, $TC(Z|X_i) = 0$. This constraint picks out models of type (b) over (a) in Fig. 1. Using the results of Eq. 1, we see that $TC(Z|X_i) = 0 \rightarrow p(x_i|z) \propto p(x_i) \prod_j p(z_j|x_i)$. The mean of this distribution is

$$\nu_{X_i|Z} = \frac{1}{1 + r_i} \sum_{j=1}^{m} \frac{R_{j,i}}{1 - R^2_{j,i}} Z_j$$

with $R_{j,i} = \frac{\langle X_i Z_j \rangle}{\langle X_i^2 \rangle \langle Z_j^2 \rangle}$, $r_i = \sum_{j=1}^{m} \frac{R^2_{j,i}}{1 - R^2_{j,i}}$.

(6)

If we replace $\mu_{X_i|Z}$ with $\nu_{X_i|Z}$ in the expression above, we get an upper bound on our original objective that becomes tight exactly when $TC(Z|X_i) = 0$.

$$\min_{Z_j|X \sim \mathcal{N}(W_j x, \eta^2)} \sum_{i=1}^{p} \frac{1}{2} \log \langle (X_i - \nu_{X_i|Z})^2 \rangle + \sum_{j=1}^{m} \frac{1}{2} \log \langle Z_j^2 \rangle$$

(7)

If $X$ is Gaussian then choosing $Z_j|X$ to be Gaussian ensures the joint distribution is also Gaussian. This objective depends only on pairwise statistics and requires no matrix inversion. The global minimum is achieved for non-overlapping Gaussian latent factor models.

Quasi-Newton optimization. We derived a quasi-Newton optimization procedure for the optimization in Eq. 7. However, unlike gradient descent, exhibits super-linear rates of convergence. Details are presented in Appendix B but sketched here. After a nonlinear change of variables to $R_{j,i}$, we are able to compute the gradient and Hessian. Then we apply the coordinate steps in the original coordinate space, $W$. The gradient (with respect to $R_{j,i}$), $G_{j,i}$, is as follows.

$$G_{j,i} = \frac{(1 + R^2_{j,i}) Q_{j,i} - 2 R_{j,i} r_i}{(1 - R^2_{j,i})^2 (1 + q_i - r_i^2)} - 2 R_{j,i} \frac{R_{j,i}}{(1 - R^2_{j,i})^2 (1 + r_i) + (HW)_{j,i}}$$

(8)
Besides the quantities in Eq. [6] we introduced the following quantities.

\[ M_{j,k} = \frac{\langle Z_j Z_k \rangle - \delta_{j,k}}{\sqrt{\langle Z_j^2 \rangle \langle Z_k^2 \rangle}}, \quad Q_{j,i} = \sum_{k=1}^{m} \frac{M_{j,k} R_{k,i}}{(1 - R_{k,i}^2)}, \quad q_i = \sum_{j=1}^{m} \frac{Q_{j,i} R_{j,i}}{(1 - R_{j,i}^2)} \]

\[ H_{j,k} = \delta_{j,k} \sqrt{\langle Z_j^2 \rangle} + \frac{1 - \delta_{j,k}}{\sqrt{\langle Z_j^2 \rangle \langle Z_k^2 \rangle}} \sum_{i=1}^{p} \frac{R_{j,i} R_{k,i}}{(1 - R_{j,i}^2)(1 - R_{k,i}^2)(1 + q_i - r_i^2)} \]

The main terms in the Hessian are diagonal plus rank one terms that can be inverted analytically. Then we get a quasi-Newton update step for \( R \) of the form \( R^{t+1} = R^t - \alpha \Delta \), where \( \Delta \) is the approximate inverse Hessian times the gradient. We can multiply this update rule by a constant matrix, \( \Lambda \), to get an update for the variable, \( U_{j,i} \equiv (\Lambda R)_{j,i} \). The quasi-Newton update for \( U \) is as follows.

\[ \Delta_{U_{j,i}} = \frac{G_{j,i}}{\sqrt{\langle Z_j^2 \rangle}} - \frac{(RG^\top)_{j,j} W_{j,i}}{\langle Z_j^2 \rangle - 1/2} \]  

Finally, we want to recover the weights, \( W \), from our update of \( U \) which is done through the relation, \( W_{j,i} = U_{j,i}/(U_j \Sigma U_j^\top) \). The objective is non-convex so this update only guarantees convergence to a local optimum. We use backtracking to set \( \alpha \in (0, 1] \) to ensure updates obey the Wolfe conditions. We show convergence results in Fig. 3.

**Annealing** In empirical evaluations, we were surprised to see that this update worked better for noisy data than for nearly deterministic latent factor models. We hypothesize that noisier latent factor models exhibit stronger curvature in the optimization space. Therefore, we implemented an annealing procedure to improve results for nearly deterministic factor models. We replace the covariance matrix appearing in Eq. [9] with a noisy version, \( \Sigma_\epsilon = (1 - \epsilon) \Sigma + \epsilon I \), for some \( \epsilon \). Then we train to convergence, reduce \( \epsilon \), and use the previous weight matrix as the initialization for the next step. We used an annealing schedule for \( \epsilon \) of \([0.6, 0.6^2, 0.6^3, 0.6^4, 0.6^5, 0.0]\) in all experiments.

**Complexity** The computational complexity of our method is dominated by matrix multiplications of an \( m \times p \) weight matrix and a \( p \times n \) data matrix, giving a computational complexity of \( O(mp) \). This is only linear in the number of variables making it an attractive alternative to standard methods that are at least quadratic like PCA or GLASSO. Note that although the covariance matrix appears in our results, we never have to construct it since terms like \( W \Sigma W^\top = WX X^\top W^\top = (WX)(WX)^\top \) can be calculated using \( 2mpn + m^2n \) operations from the raw data. The solution depends only on marginals of the form \( \langle X_j Z_j \rangle \) whose estimation error does not depend on the dimension, \( p \).

---

**Data:** Data matrix, \( n \) iid samples of vectors, \( \mathbf{X} \in \mathbb{R}^p \)

**Result:** Weight matrix, \( W \), optimizing Eq. [7]

Subtract mean and scale each column of data;

\( W_{j,i} \sim \mathcal{N}(0, 1/\sqrt{p}) \);

for \( \epsilon \) in annealing schedule do

while not converged do

  Calculate \( \Delta U \) using \( \Sigma_\epsilon \) in Eq. [6];

  Backtrack to pick \( \alpha \in (0, 1] \) satisfying Wolfe conditions;

  \( U \leftarrow U - \alpha \Delta U \);

  \( W_{j,i} \leftarrow U_{j,i}/(U_j \Sigma U_j^\top) \);

end

**Algorithm 1:** Implementation is available at [http://github.com/gregversteeg/LinearCorEx](http://github.com/gregversteeg/LinearCorEx)

---

**Figure 3:** Relative error of the objective after each iteration compared to the converged value.
5 Results

5.1 Evidence for a blessing of dimensionality in structure recovery

The lower bound on sample complexity that we derived for non-overlapping Gaussian latent factor models suggests that structure recovery should get easier as we add more variables. Do any methods exhibit this desirable property? To test this claim, we generate synthetic data from the model in Sec. 2 (a NGLF model with \( m \) sources, \( p \) variables, \( n \) samples, and equal groups of size \( p/m \)) and then use various methods to recover the structure of the latent factor model.

Recovering the structure just corresponds to correctly clustering the observed variables, so we consider various clustering approaches as baselines. For decomposition approaches like factor analysis (FA), non-negative matrix factorization (NMF), principal component analysis (PCA), and independent component analysis (ICA), we cluster variables according to the latent factor whose weight has the maximum magnitude for a variable. Other clustering methods include k-means, hierarchical agglomerative clustering using Euclidean distance and the Ward linkage rule (Hier.), and spectral clustering (Spec.). It is well known that factor analysis suffers from an unidentifiability problem because the latent factors can be arbitrarily rotated without changing the results [25]. To rectify this we include factor analysis with the Varimax rotation (FA+V) [16] to find more meaningful clusters. Finally, we consider the latent tree reconstruction method [7], where clusters of variables are identified through the “information distance” (ID). We measure the quality of clusters using normalized mutual information which varies between 0 for random clusters and 1 for perfect clusters.

We show an example result in Fig. 4 with varying \( p \) and \( n = 300, m = 64, SNR = 0.1 \). Thm. 2.1 forbids perfect recovery (NMI=1) for any method if the dimensionality \( p < 584 \). For a fixed number of samples, increasing the dimensionality seems to benefit our proposed method but no others. In other words, ours is the only method to see a blessing of dimensionality. Note, however, that there is still a large gap between empirical results and our lower bound which would allow perfect recovery to the right of the arrow. It is unknown whether the lower bound is too loose, or our method can be improved, or both. Note that the next best approach is based on information distance (ID) [7] and it is natural to wonder if it might eventually improve with more variables. Because calculating information distances has quadratic time and space complexity, it is difficult to answer this question.

5.2 Covariance estimation

The structure recovery setting was somewhat artificial to facilitate comparison with our theoretical results. We now consider the more difficult and useful problem of estimating the strength of relationships in Gaussian graphical models, i.e., covariance matrix estimation. A formula for estimating \( \Sigma \) from learned weights, \( W \), is provided in Sec. B.3.

We estimate the covariance matrix from training data using a variety of approaches. To measure the quality of each covariance matrix estimate, we evaluate the negative log-likelihood on test data [25]. We include the empirical covariance matrix as a baseline but it is well known that in the undersampled regime, \( n < p \), the estimate will be ill-conditioned and perform poorly. As a simple and
robust baseline, we include the diagonal covariance matrix where we put the sample variances on the diagonal. Failing to beat the simple independent baseline signals poor covariance estimation.

We compare with several classes of covariance estimators. Ledoit and Wolf (henceforth LW) introduced a simple approach to correct the deficiencies of the empirical estimate. Essentially, they take as a prior that the variables are independent and then give a Bayesian estimate of the covariance given the data. This type of “Bayesian shrinkage estimator” will recover the empirical covariance in the large $n$ limit while regularizing the estimate towards independence when $n \ll p$ \cite{17}. A large and growing literature on sparse, inverse covariance estimation achieves regularization by assuming that the underlying graphical model (or inverse covariance matrix, a.k.a. precision matrix) of the variables is sparse \cite{14,15,20,4,22}. We include the most popular variant, the Graphical LASSO or GLASSO \cite{12}. For GLASSO we used cross validation with an iteratively refined grid to pick hyper-parameters. Finally, we also include regular factor analysis \cite{5,25}, a maximum likelihood approach to modeling the covariance matrix as diagonal plus low rank.

**Synthetic results** We first evaluate covariance estimation on synthetic data where the ground truth covariance matrix is known. In particular, the NGLF model leads to covariance matrices where each block is diagonal plus rank one (other entries are zero). We set $p = 128$, $SNR = 5$ and varied the number of samples used for estimation. We consider the case of large groups with 64 variables in each block (a) and small groups with 8 variables in each block (b). We show mean and standard deviation over five random runs for each point. The results are shown in Fig. 5. The best possible performance is given by the ground truth (GT) line. The empirical covariance estimate fails when $n \leq p$ (as expected) and factor analysis is also not very competitive. LW nicely interpolates between the empirical covariance in the large $n$ limit and the independent baseline when $n$ is small. Our method clearly outperforms all others in Fig. 5(a). The covariance matrix in Fig. 5(b) was chosen to have a sparser structure to tailor the problem for GLASSO. Our method still outperforms GLASSO over most of the range, losing narrowly for $n \leq 16$ samples.
**Stock market data** For a high-dimensional real world dataset we look at stock market data. The covariance matrix plays a central role for estimating risk and this has motivated much development in covariance estimation [17]. Because the stock market is highly non-stationary, it is desirable to estimate covariance using only a small number of samples consisting of the most recent data. We considered the weekly percentage returns for U.S. stocks from January 2000 to January 2017 freely available from [http://quandl.com](http://quandl.com). After excluding stocks that did not have returns over the entire period, we were left with 1491 companies. We trained on \( n \) weeks of data to learn a covariance matrix using various methods then evaluated the negative log likelihood on the subsequent 26 weeks of test data. Each point in Fig. 6 is an average from rolling the training and testing sets over the entire time period. For component-based methods (PCA, FA, our method) we used 30 components.

![Figure 6: Covariance matrices learned from stock market data are evaluated according to negative log-likelihood (lower is better) on test data. The points for the empirical covariance and most of Ledoit-Wolf are above the top of the y axis.](image)

The empirical covariance matrix is highly under-sampled so it is not surprising that it performs poorly. Ledoit-Wolf does not help much in this regime, doing worse than the independent baseline and PCA. With enough samples, factor analysis is able to beat the independent baseline. Because GLASSO looks for sparse solutions, it is able to consistently match or beat the independent baseline (which can be interpreted as the maximally sparse solution). Our method consistently outperforms all the other methods. The stock market is not well modeled by sparsity, but attributing correlations to a small number of latent factors appears to be effective. Our approach leverages the high-dimensional data more efficiently than standard factor analysis. We visualize some latent factors in Appendix C and find a close correspondence with industry sectors.

### 6 Related work

A notable omission in our results is a class of latent factor models that can be cast as convex optimization problems [6, 21]. While these methods are rightly celebrated, their polynomial computational complexity often makes them intractable for high-dimensional problems like the ones considered here. The largest example used in [6] had 84 variables and 5 latent factors, while we considered problems with thousands of variables. Whether a blessing of dimensionality could be observed for this approach is an open question. Latent tree reconstruction methods, while computationally intensive, were tractable and a clear runner-up in the structure recovery experiments [7].

While sparse methods and tractable approximations have enjoyed a great deal of attention [12, 20, 14, 13, 4, 19, 22], marginalizing over a latent factor model does not necessarily lead to a sparse model over the observed variables. Many highly correlated systems, like the stock market, seem better modeled through a small number of latent factors. Factor methods have a long history in finance, though the emphasis is on using a small set of known factors, see [11] and references therein. Deficiencies of standard latent factor methods, like unidentifiability, may have muted the apparent usefulness of this class of methods [25].

Learning through optimization of information-theoretic objectives has a long history focusing on mutual information [18, 3, 26]. Minimizing \( TC(Z) \) is well known as ICA [8, 15]. The problem of minimizing \( TC(X|Z) \) is less known but related to the Wyner common information [32] and has also been recently investigated as an optimization problem [23, 29]. A similar approach was used to recover discrete latent factors to lower bound \( TC(X) \) [27].
7 Conclusion

For some types of big data, it is easier to increase the number of variables than the number of samples. It may be that we collect in-depth data on a small number of individuals (clinical or social science studies) or we want to understand a high-dimensional system under specific circumstances (i.e., what is the covariance of stocks over the last month?). Few computational approaches benefit from increasing the number of variables with a fixed, small number of samples. In this paper, we presented theoretical and experimental evidence that a special class of latent factor models benefit from dimensionality. Our approach is based on an information-theoretic measure that can be sensibly optimized without making assumptions about the true data generating model, providing useful results even for messy, real-world data like the stock market.

Acknowledgments

We thank Andrey Lokhov, Marc Vuffray, and Seyoung Yun for valuable conversations about this work. GV acknowledges support from DARPA grant W911NF-16-1-0575. This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] S.-i. Amari and H. Nakahara. Correlation and independence in the neural code. Neural computation, 18(6):1259–1267, 2006.
[2] M.-F. Balcan, Y. Liang, D. P. Woodruff, and H. Zhang. Optimal sample complexity for matrix completion and related problems via l2-regularization. arXiv preprint arXiv:1704.08683, 2017.
[3] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural computation, 7(6):1129–1159, 1995.
[4] T. Cai, W. Liu, and X. Luo. A constrained l1 minimization approach to sparse precision matrix estimation. Journal of the American Statistical Association, 106(494):594–607, 2011.
[5] R. B. Cattell. Factor analysis: an introduction and manual for the psychologist and social scientist. 1952.
[6] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pages 1610–1613. IEEE, 2010.
[7] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky. Learning latent tree graphical models. The Journal of Machine Learning Research, 12:1771–1812, 2011.
[8] P. Comon. Independent component analysis, a new concept? Signal processing, 36(3):287–314, 1994.
[9] T. M. Cover and J. A. Thomas. Elements of information theory. Wiley-Interscience, 2006.
[10] D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. AMS Math Challenges Lecture, 1:32, 2000.
[11] J. Fan, Y. Fan, and J. Lv. High dimensional covariance matrix estimation using a factor model. Journal of Econometrics, 147(1):186–197, 2008.
[12] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432–441, 2008.
[13] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In Advances in Neural Information Processing Systems, pages 3165–3173, 2013.
[14] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Quic: quadratic approximation for sparse inverse covariance estimation. Journal of Machine Learning Research, 15(1):2911–2947, 2014.
[15] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

[16] H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

[17] O. Ledoit and M. Wolf. Honey, i shrunk the sample covariance matrix. *UPF Economics and Business Working Paper*, (691), 2003.

[18] R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[19] Y. Liu and A. Willsky. Learning gaussian graphical models with observed or latent variables. In *Advances in Neural Information Processing Systems*, pages 1833–1841, 2013.

[20] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

[21] Z. Meng, B. Eriksson, and A. Hero. Learning latent variable gaussian graphical models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1269–1277, 2014.

[22] S. Misra, M. Vuffray, A. Y. Lokhov, and M. Chertkov. Towards optimal sparse inverse covariance selection through non-convex optimization. arXiv preprint arXiv:1703.04886, 2017.

[23] G. Op’t Veld and M. C. Gastpar. Caching gaussians: Minimizing total correlation on the gray–wyner network. In *Proceedings of the 50th Annual Conference on Information Systems and Sciences (CISS)*, 2016.

[24] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, NY, NY, USA, 2009. ISBN 052189560X, 9780521895606. URL http://portal.acm.org/citation.cfm?id=1642718.

[25] C. Shalizi. Advanced data analysis from an elementary point of view, 2013.

[26] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. arXiv:physics/0004057, 2000.

[27] G. Ver Steeg and A. Galstyan. Discovering structure in high-dimensional data through correlation explanation. *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[28] G. Ver Steeg and A. Galstyan. Maximally informative hierarchical representations of high-dimensional data. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[29] G. Ver Steeg and A. Galstyan. The information sieve. In *International Conference on Machine Learning (ICML)*, 2016.

[30] W. Wang, M. J. Wainwright, and K. Ramchandran. Information-theoretic bounds on model selection for gaussian markov random fields. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1373–1377. IEEE, 2010.

[31] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[32] A. D. Wyner. The common information of two dependent random variables. *Information Theory, IEEE Transactions on*, 21(2):163–179, 1975.
A Proofs

A.1 Proof of Thm. 2.1

Proof. Consider the class of NGLF models for which we have \( p \) variables and \( m \) latent factors and each latent factor has exactly \( p/m \) children in the observed variables. To distinguish the structure among this class of models corresponds to partitioning the observed variables into \( m \) equally sized groups. The number of ensembles is,

\[
M = \binom{p}{p/m, \ldots, p/m} \frac{1}{m!},
\]

the multinomial coefficient for dividing \( p \) items into \( m \) equally sized boxes, divided by the number of indistinguishable permutations among boxes, \( m! \). We take \( \theta \in \{1, \ldots, M\} \) to be an index specifying a model in this ensemble. The \( Z_j \)'s are independent Gaussian variables with variance, \( b \), and each variable \( X_i = Z_{pa_\theta(i)} + \eta_i \), where the parent of \( X_i \) in model \( \theta \) is \( pa_\theta(i) \) and \( \eta_i \) is independent noise with variance \( a \). We can write the covariance matrix over observed variables,

\[
\Sigma_{\theta,i,j} = \langle X_i X_j \rangle = b \delta_{pa_\theta(i),pa_\theta(j)} + a \delta_{i,j},
\]

where \( \delta \) is the Kronecker delta.

Fano’s inequality tells us that the probability of an error, \( \epsilon \), in picking the correct index, \( \theta \), given \( n \) samples of data, \( X_1^{1:p} \) is bounded as follows.

\[
\epsilon \geq 1 - \frac{I(\theta; X_1^{1:n}) - 1}{\log M}
\]

Following [30], we use an upper bound for the mutual information,

\[
I(\theta; X_1^{1:n}) \leq \frac{n}{2F},
\]

where \( F = \log \det \bar{\Sigma} - \frac{1}{M} \sum_\theta \log \det \Sigma_\theta \)

and \( \bar{\Sigma} = 1/M \sum_\theta \Sigma_\theta \). Re-arranging Fano’s inequality gives the following sample complexity bound.

\[
n \geq 2 \frac{(1 - \epsilon) \log M - 1}{F} \tag{10}
\]

All that remains is to find an expression for \( F \). To build intuition, we explicitly write out the case for \( p = 4, m = 2 \), for some \( \theta \).

\[
\Sigma_\theta = \begin{bmatrix}
b + a & b & 0 & 0 \\
b & b + a & 0 & 0 \\
0 & 0 & b + a & b \\
0 & 0 & b & b + a
\end{bmatrix} \tag{11}
\]

Clearly this is a block diagonal matrix where each block is a diagonal plus rank one (DPR1) matrix. After we average over all \( \theta \) to get \( \bar{\Sigma} \), every off-diagonal entry will be the same, equal to the probability of \( j \neq i \) being in the same group as \( i \), or \( (p/m - 1)/(p - 1) \). Therefore \( \bar{\Sigma} \) is also a DPR1 matrix. Using standard identities for block diagonal and DPR1 matrices, we calculate the determinants.

\[
\det \Sigma_\theta = a^p \left( 1 + \frac{b}{a} \frac{p}{m} \right)^m
\]

\[
\det \bar{\Sigma} = a^p \left( 1 + \frac{b}{a} \frac{p}{m} \right) \left( 1 + \frac{b}{a} \frac{m}{m - 1} \right)^{p-1} \tag{12}
\]

Finally we can combine all of these expressions to get a lower bound for sample complexity that depends only on \( p, m \), and the signal-to-noise ratio, \( \text{SNR} = b/a \).

A.2 Proof of Thm. 3.1

Proof. Because \( TC \) is always non-negative,

\[
TC(X|Z) + TC(Z) = 0 \iff TC(Z) = 0 \text{ and } TC(X|Z) = 0. \tag{13}
\]

We also have the following standard statements [9].

\[
TC(X|Z) = 0 \iff \forall x, z, \ p(x|z) = \prod_{i=1}^{n} p(x_i|z)p(z) \tag{14}
\]
We start by learning a linear function

\[
TC(Z) = 0 \iff \forall z, \quad p(z) = \prod_{j=1}^{m} p(z_j)
\]  

(15)

Putting these together, we have

\[
\forall x, z, \quad p(x, z) = \prod_{i=1}^{n} \prod_{j=1}^{m} p(x_i|z)p(z_j).
\]  

(16)

We can see that this statement is equivalent to the definition of a Bayesian network for random variables \(X, Z\) with respect to the graph in Fig.1.

A.3 Proof of Thm. 3.2

Proof. First we show that the non-overlapping latent factor graphical model implies the constraints are satisfied. Thm. 3.1 establishes that the model implies \(TC(X|Z) + TC(Z) = 0\). We must show that the additional restriction that each \(X_i\) has only one parent, \(Z_j\), implies the condition \(\forall i, TC(Z|X_i) = 0\). Looking at the rules for d-separation [24], if any \(X_i\) has more than one parent, \(Z_j, Z_k \neq j\), then conditioning on \(X_i\) (a “head-to-head” node) unblocks the path between the two parents so that \(Z_j, Z_k\) are no longer independent after conditioning on \(X_i\). Therefore, if there is any overlap in parents for node \(X_i\), then \(TC(Z|X_i) > 0\).

Now, we show that for Gaussian distributions the constraints, \(TC(X|Z) + TC(Z) = 0, \forall i, TC(Z|X_i) = 0\) implies the non-overlapping latent factor graphical model.

\[
\forall x, z, \quad p(x, z) = \prod_{i} p(x_i|z) \prod_{j} p(z_j)
\]  

(17)

To get the non-overlapping latent factor decomposition, we have to show that \(TC(Z|X_i) \rightarrow p(x_i|z) = p(x_i|z_j)\) for only a single \(j\).

\[
TC(Z|X_i) = 0 \rightarrow \forall x_i, z, \quad p(z|x_i) = \prod_{j} p(z_j|x_i)
\]

Now we re-arrange this expression.

\[
p(x_i|z) = p(x_i)/p(z) \prod_{j} p(z_j|x_i) = p(x_i) \prod_{j} p(z_j|x_i)/p(z_j) = p(x_i) \prod_{j} p(z_j, x_i)/(p(x_i)p(z_j))
\]  

(18)

We must show that \(X_i\) has at most one parent in \(Z\). For Gaussian distributions, \(TC(Z) = 0\) iff the covariance matrix, \(\Sigma^{Z}\), is zero on the off-diagonals. The conditional covariance has a well-known formula, \(\Sigma^{Z|X_i} = \Sigma^{Z} - \langle X_i Z_j \rangle \langle X_i Z_k \rangle / \langle X_i^2 \rangle\). Since \(\Sigma^{Z}\) is zero on the off-diagonals, \(TC(Z|X_i)\) implies that for each pair, \(j, k \neq j\), either \(\langle X_i Z_j \rangle = 0\) or \(\langle X_i Z_k \rangle = 0\). Therefore, for all but at most one index, \(j\), it must be the covariance of \(X_i\) and \(Z_j\) is zero, so that \(p(z_j, x_i) = p(x_i)p(z_j)\). Putting this in Eq. [18] we get \(p(x_i|z) = p(x_i|z_k)\), completing the proof.

B Derivation of optimization procedure

We start by learning a linear function \(z_j = \sum_i W_{j,i} x_i + \epsilon_j\). The added noise \(\epsilon_j \sim \mathcal{N}(0, \eta^2)\) sets the scale of the latent factors. If we achieved the global minimum of zero for this objective, then our equivalence theorem tells us that we have reconstructed a perfect generative model of the data. This
objective can be re-written.
\[
TC(X|Z) + TC(Z) = \sum_{i=1}^{p} H(X_i|Z) - H(X|Z) + \sum_{j=1}^{m} H(Z_j) - H(Z)
\]
\[
= \sum_{i=1}^{p} H(X_i|Z) + \sum_{j=1}^{m} H(Z_j) - (H(X|Z) + H(Z))
\]
\[
= \sum_{i=1}^{p} H(X_i|Z) + \sum_{j=1}^{m} H(Z_j) - (H(Z|X) + H(X))
\]
\[
= \sum_{i=1}^{p} H(X_i|Z) + \sum_{j=1}^{m} H(Z_j) + TC(X) - \sum_{i} H(X_i) - \sum_{j} H(Z_j|X)
\]
\[
= \sum_{i=1}^{p} H(X_i|Z) + \sum_{j=1}^{m} H(Z_j) + TC(X) - \sum_{i} H(X_i) - m/2 \log \eta^2
\]
\[\propto \sum_{i=1}^{p} H(X_i|Z) + \sum_{j=1}^{m} H(Z_j) \]

The first two lines invoke definitions and re-arrange. The third line uses Bayes’ rule to rewrite the entropies. The fourth line adds and subtracts \(H(X_i)\) to get a TC term, and also invokes conditional independence of \(Z\)’s conditioned on \(X\). Finally, we ignore constants for optimization purposes, but we calculate them because of the following useful bound.
\[
TC(X|Z) + TC(Z) \geq 0 \rightarrow
\]
\[
TC(X) \geq \sum_{i=1}^{p}(H(X_i) - H(X_i|Z)) - \sum_{j=1}^{m} H(Z_j) + m/2 \log \eta^2
\]
Optimizing \(Z\) leads to successively better lower bounds on \(TC(X)\) \[28\].

B.1 Gradient

We are optimizing the following objective.
\[
\min_{Z_j|X \sim N(W_j, \nu^2, \eta^2)} \sum_{i=1}^{p} 1/2 \log \langle (X_i - \nu_{X_i}Z_j)^2 \rangle + \sum_{j=1}^{m} 1/2 \log \langle Z_j^2 \rangle
\]
(20)

Where we have,
\[
\nu_{X_i}Z_j = \frac{1}{1 + r_i} \sum_{j=1}^{m} \frac{R_{j,i}}{1 - R_{j,i}^2} Z_j, \quad R_{j,i} = \frac{\langle X_i Z_j \rangle}{\sqrt{\langle X_i^2 \rangle \langle Z_j^2 \rangle}}, \quad r_i = \sum_{j=1}^{m} \frac{R_{j,i}^2}{1 - R_{j,i}^2}.
\]
(21)

Note that the objective and constraints are invariant to the scale of \(Z_j\). Therefore any solution with matrix \(W\) and noise \(\eta^2\) can be scaled to an equivalent solution with a scaled \(W\) and \(\eta^2 = 1\). Therefore we set \(\eta^2 = 1\) for the remainder of the derivation.

Our next step is to make a change of variables from \(W\) to \(R\). We will derive an update in \(R\)-space and then map the solution back to \(W\). We summarize some of the relevant identities. We will use \(i, \ell\) for indices that run \(1, \ldots, p\) and \(j, k\) for indices that run \(1, \ldots, m\). The covariance matrix over \(X\) is \(\Sigma\) and its inverse is \(\Lambda\). We assume that the data has been standardized to have \(\langle X_i \rangle = 0, \langle X_i^2 \rangle = 1\). Recall that \(Z_j = \sum_i W_{j,i} X_i + \epsilon_j\) where \(\epsilon_j\) is a \(N(0, 1)\) random variable that is independent of \(X, Z\).
\[
\langle Z_j Z_k \rangle = \sum_{i, \ell} W_{j,i} W_{j,\ell} \Sigma_{i, \ell} + \delta_{j,k}, \quad \langle X_i Z_j \rangle = \sum_{\ell} W_{j,i} \Sigma_{i, \ell},
\]
(22)

As always, \(\delta_{j,k}\) is the discrete delta (one if indices match or zero otherwise). Clearly, \(R\) can be written in terms of \(W\), but now we show how to write \(W\) in terms of \(R\).
\[
W_{j,i} = \frac{\sum_{\ell} \Lambda_{i, \ell} R_{j, \ell}}{\sqrt{1 - \sum_{\ell, \ell'} R_{j,\ell} \Lambda_{\ell, \ell'} R_{j, \ell'}}}
\]
(23)
Another useful identity is that \( \langle Z_j^2 \rangle = 1/(1 - \sum_{i,e} R_j^e \Lambda_{i,e} R_j^e) \). We re-express our objective, \( O \) from Eq. 7 in terms of \( R \).

\[
O = \sum_i 1/2 \log(1 + q_i - r_i^2) - \log(1 + r_i) + \sum_j 1/2 \log \langle Z_j^2 \rangle \tag{24}
\]

We defined \( r_i \) in terms of \( R \) and \( q_i \) can be defined as well.

\[
M_{j,k} = \frac{(Z_j Z_k) - \delta_{j,k}}{\sqrt{(Z_j Z_k)}} = \sum_{i,\ell} R_{j,i} \Lambda_{i,\ell} R_{k,\ell}, Q_{j,i} = \sum_{k=1}^m M_{j,k} R_{k,i}, q_i = \sum_{j=1}^m Q_{j,i} R_{j,i} \tag{25}
\]

Next, we can take the derivatives with respect to \( R_{j,i} \) to get the following.

\[
G_{j,i} = \frac{\partial O}{\partial R_{j,i}} \approx \frac{(1 + R_{j,i}^2) Q_{j,i} - 2 R_{j,i} r_i}{(1 - R_{j,i}^2)^2 (1 + q_i - r_i^2)} - 2 \frac{R_{j,i}}{(1 - R_{j,i}^2)^2 (1 + r_i)} + \sum_{\ell} \Lambda_{i,\ell} R_{j,\ell} R_{k,\ell} + \sum_{k=1}^p \sum_{\ell=1}^p W_{k,\ell} \frac{1 - \delta_{j,k}}{\sqrt{(Z_j Z_k)}} \frac{R_{j,i} R_{k,i} (1 - R_{j,i}^2)(1 - R_{k,i}^2)(1 + q_i - r_i^2)}{(1 - R_{j,i}^2)^2 (1 + q_i - r_i^2)}
\]

In the paper text, we translated the last two terms back into expressions in terms of \( W \), using \( H \).

### B.2 Hessian

Consider just the second to last line in the gradient above. Taking the derivative of this term with respect to \( R_{j,i} \) gives the following.

\[
\frac{\partial^2 O}{\partial R_{j,i} \partial R_{k,\ell}} \approx \frac{\delta_{j,k} \Lambda_{i,\ell}}{1 - \sum_{\ell} \sum_{\ell'} R_{j,\ell} \Lambda_{i,\ell'} R_{j,\ell'} R_{k,\ell'}^2} \frac{2 \delta_{j,k} \sum_{\ell} \Lambda_{i,\ell} R_{j,\ell} \sum_{\ell'} \Lambda_{i,\ell'} R_{j,\ell'}}{(1 - \sum_{\ell} \sum_{\ell'} R_{j,\ell} \Lambda_{i,\ell'} R_{j,\ell'}^2)^2}
\]

This is a diagonal plus rank one update, so the matrix inverse can be calculated.

\[
\left( \frac{\partial^2 O}{\partial R_{j,i} \partial R_{k,\ell}} \right)^{-1} \approx \left(1 - \sum_{\ell} R_{j,\ell} \Lambda_{i,\ell} R_{j,\ell} R_{k,\ell}^2 \right) (\Sigma_{i,\ell})^{-1} - \frac{2 R_{j,i} R_{k,\ell}}{1 + \sum_{\ell} R_{j,\ell} \Lambda_{i,\ell} R_{j,\ell} R_{k,\ell}}
\]

We apply this to the gradient, giving us a quasi-Newton update step for \( R \) of the following form, where \( \Delta \) is the approximate inverse Hessian times the gradient.

\[
R_{j,i}^{t+1} = R_{j,i}^t - \alpha \Delta_{j,i}
\]

We can multiply this update rule by a constant matrix, \( \Lambda \), to get an update for the variable, \( U_{j,i} = (RA)_{j,i} \). The quasi-Newton update for \( U \) is as follows.

\[
\Delta_{U_{j,i}} = \frac{G_{j,i}}{\sqrt{(Z_j Z_k)}} - \frac{(RG^\top)_{j,j} W_{j,i}}{(Z_j Z_k) - 1/2} \tag{26}
\]

Finally, we want to recover the weights, \( W \), from our update of \( U \) which is done through the relation, \( W_{j,i} = U_{j,i}/(U_j \Sigma U_j^\top) \).

### B.3 Estimating the covariance

A method for estimating the covariance is as follows. First, we have assumed that the data is scaled so that \( \langle X_i^2 \rangle = 1, \langle X_i \rangle = 0 \), so we just need to calculate the off-diagonal terms. If \( TC(X|Z) = 0 \), this implies the conditional covariance of \( X \) given \( Z \) is diagonal. However, we can also write the conditional covariance as follows.

\[
cov(X_i, X_{\ell}\neq i|Z) = \langle X_i X_{\ell} \rangle - \langle \mu_{X_i|Z} \mu_{X_{\ell}|Z} \rangle = 0
\]
If we assume the constraints $\forall i, TC(Z|X_i) = 0$ are satisfied, we saw in Eq. 6 that this implies $\mu_{X_i|Z} = \nu_{X_i|Z}$, where,

$$\nu_{X_i|Z} = \frac{1}{1 + r_i} \sum_{j=1}^m \frac{R_{j,i}}{1 - R_{j,i}^2} Z_j.$$ 

If also assume that $TC(Z) = 0$ so that $\langle Z_j Z_k \rangle = \delta_{j,k} \langle Z_i^2 \rangle$, then the off-diagonal elements of $\langle X_i X_\ell \rangle$ can be written as:

$$\langle X_i X_\ell \rangle = \langle \nu_{X_i|Z} \nu_{X_\ell|Z} \rangle = \frac{1}{(1 + r_i)(1 + r_\ell)} \sum_{j=1}^m \frac{R_{j,i} R_{j,\ell} \langle Z_j^2 \rangle}{(1 - R_{j,i}^2)(1 - R_{j,\ell}^2)}$$

## C Visualizing some latent factors in stock market data

We visualize learned latent factors in Fig. C.1. In this experiment, we used weekly returns from January 2014 to January 2017 for training. This means we used only 156 samples and 1491 variables (stocks). For each factor, we use the mutual information between a latent factor and stock to rank the top stocks related to a factor. For each latent factor, we sort the weeks according to high and low values of that latent factor. This allows us to see on the heatmap in Fig. C.1 that the returns for stocks associated with latent factor 0 are all high (dark color) when the latent factor is high and low (light color) when the latent factor 0 is low. The heatmaps make it clear that groups of stocks associated to the same latent factor are indeed related. Factor 0 appears to be not just banking related, but more specifically bank holding companies. Factor 5 has remarkably homogeneous correlations and consists of energy companies. Factor 9 is specific to home construction.
Figure C.1: We show 10 latent factors from a model trained on data from January 2014 to January 2017. For each factor, we show the top ten stocks that have highest mutual information with a latent factor. Colors correspond to standard deviation of returns for a given stock compared to its mean. Each column represents a trading week, but in each plot the weeks have been sorted according to the learned latent factor.