Inference of Sparse Networks with Unobserved Variables. Application to Gene Regulatory Networks

Nikolai Slavov
Princeton University, NJ 08544
nslavov@princeton.edu

Abstract

Networks are becoming a unifying framework for modeling complex systems and network inference problems are frequently encountered in many fields. Here, I develop and apply a generative approach to network inference (RC\textsubscript{web}) for the case when the network is sparse and the latent (not observed) variables affect the observed ones. From all possible factor analysis (FA) decompositions explaining the variance in the data, RC\textsubscript{web} selects the FA decomposition that is consistent with a sparse underlying network. The sparsity constraint is imposed by a novel method that significantly outperforms (in terms of accuracy, robustness to noise, complexity scaling and computational efficiency) Bayesian methods and MLE methods using $\ell_1$ norm relaxation such as K–SVD and $\ell_1$–based sparse principle component analysis (PCA). Results from simulated models demonstrate that RC\textsubscript{web} recovers exactly the model structures for sparsity as low (as non-sparse) as 50% and with ratio of unobserved to observed variables as high as 2. RC\textsubscript{web} is robust to noise, with gradual decrease in the parameter ranges as the noise level increases.

1 INTRODUCTION

Factor analysis (FA) decompositions are useful for explaining the variance of observed variables in terms of fewer unobserved variables that may capture systematic effects and allow for low dimensional representation of the data. Yet, the interpretation of latent variables inferred by FA is fraught with problems. In fact, interpretation is not always expected and intended since FA may not have an underlying generative model. A prime difficulty with interpretation arises from the fact that any rotation of the factors and their loadings by an orthogonal matrix results in a different FA decomposition that explains the variance in the observed variables just as well. Therefore, in the absence of additional information on the latent variables and their loading, FA cannot identify a unique decomposition, much less generative relationships between latent factors and observed variables.

A frequent choice for a constraint implemented by principle component analysis (PCA) and resulting in a unique solution is that the factors are the singular vectors (and thus orthogonal to each other) of the data matrix ordered in descending order of their corresponding singular values. Yet, this choice is often motivated by computational convenience rather than by knowledge about the system that generated the data. Another type of computationally convenient constraint applied to facilitate the interpretation of FA results is sparsity, in the form of sparse Bayesian FA (West 2002, Dueck 2005, Carvalho 2008), sparse PCA (d’Aspremont 2007, Sigg 2008) and FA for gene regulatory networks (Srebro 2001, Pe’er 2002). However, papers that introduce and use sparse PCA do not consider a generative model but rather use sparsity as a convenient tool to produce interpretable factors that are linear combinations of just a few original variables. In sparse PCA, sparsity is a way to balance interpretability at the cost of slightly lower fraction of explained variance.

The algorithm described in this paper (RC\textsubscript{web}) also uses a sparse prior, but RC\textsubscript{web} explicitly considers the problem from a generative perspective. RC\textsubscript{web} asserts that there is indeed a set of hidden variables that connect to and regulate the observed variables via a sparse network. Based on that model, I derive a network structure learning approach within explicit theoretical framework. This allows to propose an approach for...
sparse FA which is conceptually and computationally different from all existing approaches such as K-SVD (Aharon 2005), sparse PCA and other LARS (Bradley 2004) based methods (Banerjee 2007, Friedman 2008). $\mathcal{RC}$ web is appropriate for analyzing data arising from any system in which the state of each observed variable is affected by a strict subset of the unobserved variables. To assign the inferred latent variables to physical factors, $\mathcal{RC}$ web needs either data from perturbation experiments or prior knowledge about the factors. This framework generalizes to non-linear interactions, which is discussed elsewhere. Furthermore, I analyze the scaling of the computational complexity of $\mathcal{RC}$ web with the number of observed and unobserved variables, as well as the parameter space where $\mathcal{RC}$ web can accurately infer network topologies and demonstrate its robustness to noise in the data.

2 DERIVATION

Consider a sparse bipartite graph $\mathcal{G} = (\mathcal{E}, \mathcal{N}, \mathcal{R})$ consisting of two sets of vertices $\mathcal{N}$ and $\mathcal{R}$ and the associated set of directed edges $\mathcal{E}$ connecting $\mathcal{R}$ to $\mathcal{N}$ vertices. Define a graphical model in which each vertex $s$ corresponds to a random variable; $\mathcal{N}$ observed random variables indexed by $\mathcal{N}$ \{x_N = \{x_s | s \in \mathcal{N}\} \} whose states are functions of $\mathcal{P}$ unobserved variables indexed by $\mathcal{R}$, $\mathcal{R} = \{x_s | s \in \mathcal{R}\}$. Since the states of $\mathcal{R}$ depend on (are regulated by) $\mathcal{R}$, I will also refer to $\mathcal{R}$ as regulators. The functional dependencies are denoted by a set of directed edges $\mathcal{E}$ so that each unobserved variable $x_i | i \in \mathcal{R}$ affects (and its vertex is thus connected to) a subset of observed variables $x_{\mathcal{R}}$, \{x_S = \{x_s | s \in \mathcal{R}\} \} of the observed variables $\mathcal{N}$. Given a dataset $\mathcal{G} \in \mathbb{R}^{M \times N}$ of $M$ configurations of the observed variables $\mathcal{N}$. $\mathcal{RC}$ web aims to infer the edges $\mathcal{E}$ and the corresponding configurations of the unobserved variables, $\mathcal{R}$. If the state of each observed variable is a linear superposition of a subset of unobserved variables, the data $\mathcal{G}$ can be modeled with a very simple *generative model* (1): The data is a product between $\mathcal{R} \in \mathbb{R}^{M \times P}$ (a matrix whose columns correspond to the unobserved variables and the rows correspond to the $\mathcal{M}$ measured configurations) and $\mathcal{C} \in \mathbb{R}^{P \times N}$, the weighted adjacency matrix of $\mathcal{G}$. The unexplained variance in the data $\mathcal{G}$ is captured by the residual $\mathcal{Y}$.

$$\mathcal{G} = \mathcal{R} \mathcal{C} + \mathcal{Y} \quad (1)$$

This decomposition of $\mathcal{G}$ into a product of two matrices can be considered to be a type of factor analysis with $\mathcal{R}$ being the factors and $\mathcal{C}$ the loadings. Even when $P \ll M$ the decomposition of $\mathcal{G}$ does not have a unique solution since $\mathcal{RC} \equiv \mathcal{R}\mathcal{C} \equiv \mathcal{R}^T \mathcal{Q} \mathcal{C} \equiv \mathcal{R}^T \mathcal{C}^*$ for any orthonormal matrix $\mathcal{Q}$. Thus the identification of a unique decomposition corresponding to the structure of $\mathcal{G}$ requires additional criteria constraining the decomposition. The assumption that $\mathcal{G}$ is sparse requires that $\mathcal{C}$ be sparse as well meaning that the state of the $i^{th}$ observed vertex $x_i | i \in \mathcal{N}$ is affected by a strict subset of the unobserved variables $x_{\mathcal{R}} = \{x_s | s \in \mathcal{R}\}$. Thus, to recover the structure of $\mathcal{G}$, $\mathcal{RC}$ web seeks to find a decomposition of $\mathcal{G}$ in which $\mathcal{C}$ is sparse. The sparsity can be introduced as a regularization with a Lagrangian multiplier $\lambda$:

$$\hat{\mathcal{C}}, \hat{\mathcal{R}} = \text{arg min}_{R,C} \|\mathcal{G} - \mathcal{RC}\|_F^2 + \lambda \|\mathcal{C}\|_0 \quad (2)$$

In the equations above and throughout the paper $\|\mathcal{C}\|_F^2 = \sum_{i,j} C_{ij}^2$ denotes entry-wise (Frobenius) norm, and the zero norm of a vector or matrix ($\|\mathcal{C}\|_0$) equals the number of non-zero elements in the array.

To infer the network topology $\mathcal{RC}$ web aims to solve the optimization problem defined by (2). Since (2) is a NP-hard combinatorial problem, the solution can be simplified significantly by relaxing the $\ell_0$ norm to $\ell_1$ norm (Bradley 2004). Then the approximated problem can be tackled with interior point methods (Banerjee 2007). As an alternative approach to $\ell_1$ approximation, I propose a novel method based on introducing a degree of freedom in the singular-value decomposition (SVD) of $\mathcal{G}$ by inserting an invertible\(^3\) matrix $\mathcal{B}$.

$$\mathcal{G} = \mathcal{USV}^T \equiv \left( \mathcal{R} (\mathcal{B}^T)^{-1} \right) (\mathcal{B}^T \mathcal{V}^T) \quad (3)$$

The prior (constraint) that $\hat{\mathcal{C}}$ is sparse determines $\mathcal{B}$ that minimizes (2), and thus a unique decomposition. The goal of introducing $\mathcal{B}$ is to reduce the combinatorial problem to one that can be solved with convex minimization. When the factors underlying the observed variance are fewer than the observations in $\mathcal{G}$ there is no need to take the full SVD; if $P$ factors are expected, only the first $P$ largest singular vectors and values from the SVD of $\mathcal{G}$ are taken in that decomposition so that $\mathcal{USV}^T$ is the matrix with rank $P$ that best approximates $\mathcal{G}$ in the sense of minimizing $\|\mathcal{G} - \mathcal{USV}^T\|_F$. Since conceivably sparse decompositions may use columns outside of the best $\ell_2$ approximation, $\mathcal{RC}$ web considers taking the first $P^*$ for $P^* > P$ singular pairs. Such expanded basis is more likely to support the optimal sparse solution and especially relevant for the case when $P$ is not known. Such choice can be easily accommodated in light of the ability of $\mathcal{RC}$ web to exclude unnecessary explanatory variables, see section 4.3.

\(^3\)B is always invertible by construction, see section 4.3 and equation (4)
Next $RC_{\text{web}}$ computes $B$ based on the requirement that $C$ is sparse for the case $N > P$. To compute $B$, one may set an optimization problem (4). Once $B$ is inferred, $\hat{R}$ and $\hat{C}$ can be computed easily, $\hat{R} = USB^{-1}$ and $\hat{C} = (VB)^T$.

$$\hat{B} = \arg\min_B ||VB||_0, \text{ so that } \det(B) > 1 \quad (4)$$

The constraint on $B$ in (4) is introduced to avoid trivial and degenerate solutions, such as $B$ being rank deficient $B$. Thus the introduction of $B$ reduces (2) to a problem (4) that is still combinatorial and might also be approximated with a more tractable problem by relaxing the $\ell_0$ norm to $\ell_1$ norm and applying heuristics (Cetin 2002, Candès 2007) to enhance the solution. I propose a new approach, $RC_{\text{web}}$, outlined in the next section.

3 $RC_{\text{web}}$

Assume that a sparse $c_i^T$ corresponding to an optimal $b_i$ (the $i^{th}$ column of $B$) form $VB_i = c_i^T$ is known. Define the set of indices corresponding to non-zero elements in $c_i^T$ with $\omega_-$ and the set of indices corresponding to zero elements in $c_i^T$ with $\omega_0$. Furthermore, define the matrix $V_{\omega_0}$ to be the matrix containing only the rows of $V$ whose indices are in $\omega_0$. If $\omega_0$ and thus $V_{\omega_0}$ are known, one can easily compute $b_i$ as the right singular vector of $V_{\omega_0}$ corresponding to the zero singular value. Since $c_i^T$ and $\omega_0$ are not known, $RC_{\text{web}}$ approximates $b_i$ (the smallest right singular vector of $V_{\omega_0}$) with $v_s$, the smallest right singular vector of $V$. This approximation relies on assuming that a low rank perturbation in a matrix results in a small change in its smallest singular vectors (Wilkinson 1988, Benachy-Georges 2009). Thus given that $RC_{\text{web}}$ is looking for the sparsest solution and the set $\omega_-$ is small relative to $N$, the angle between the singular vectors of $V_{\omega_0}$ and $V$ is small as well. Therefore, $v_s$ can serve as a reasonable first approximation of $b_i$. Then $RC_{\text{web}}$ systematically and iteratively uses and updates $v_s$ by removing rows of $V$ until $v_s$ converges to $b_i$ or equivalently $V_{\omega_0}$ becomes singular for the largest set of $\omega_0$ indices. When $V_{\omega_0}$ becomes singular, all elements of $c_i$ whose indices are in $\omega_0$ become zero.

$RC_{\text{web}}$ also has an intuitive geometrical interpretation. Consider the matrix $V$ mapping the unit sphere in $\mathbb{R}^P$ (the sphere with unit radius from $\mathbb{R}^P$) to an ellipsoid in $\mathbb{R}^N$. The axes of the ellipsoid are the left singular vectors of $V$. In this picture, starting with $\omega_- = \{\emptyset\}$ and $\omega_0 = \{1, \ldots, N\}$, solving (4) requires moving the fewest number of indices from $\omega_0$ to $\omega_-$ so that $V_{\omega_0}$ maps a vector from $\mathbb{R}^P$ to the origin. How to choose the indices to be moved? At each step $RC_{\text{web}}$ chooses $i_{max}$, the index of the largest element (by absolute value) of the smallest singular value of $V$ with the smallest singular value. $RC_{\text{web}}$ moves $i_{max}$ from $\omega_0$ to $\omega_-$ effectively selecting the dimension whose projection is easiest to eliminate and removing its largest component, which minimizes as much as possible the projection in that dimension. $RC_{\text{web}}$ keeps moving indices from $\omega_0$ to $\omega_-$ using the same procedure until the smallest right singular vector of $V_{\omega_0}$ converges to $b_i$ and the smallest singular value of $V_{\omega_0}$ approaches zero. $RC_{\text{web}}$ is guaranteed to stop after at most $(N-P+1)$ steps since after removal of $(N-P+1)$ indices from $\omega_0$, $V_{\omega_0}$ will be at most rank $P-1$. If $RC_{\text{web}}$ finds a sparse solution it will converge in fewer steps.

1. Task: $\hat{b}_i = \min_{b_i^{T}b_i \geq 1} \|VB_i\|_0$

2. Initialization:
   - $\omega_- = \{\emptyset\}$ and $\omega_0 = \{1, 2, \ldots, N\}$
   - Set $K_{\omega_0}^{-1} = (V_{\omega_0}^TV_{\omega_0})^{-1} = I \in \mathbb{R}^{P \times P}$
   - Set $J = 1$;
   - $i_{max} = \arg\max_i \{||V_i||\}$
   - $\omega_0 = \{i_{max}\}$
   - $\omega_- = \{i | i \in \omega_0, i \neq i_{max}\}$
   - Update $K_{\omega_0}^{-1} = \text{RankUpdate}(K_{\omega_0}^{-1}, V_{i_{max}})$

3. Cycle: $J = J + 1$ Repeat until convergence
   - Find the eigenvector $v$ for $K_{\omega_0}$ with the largest eigenvalue $\lambda_{max}$
   - If $\lambda^{-1}_{max} \approx 0$ or $v^J \rightarrow v^{J-1}$, $\hat{b}_i \equiv v$; STOP
   - Compute the left singular vector of $V_{\omega_0}$
     - $u = s^{-1}V_{\omega_0}$
     - $i_{max} = \arg\max_i ||u_i||$ if $\{||u_i|| \geq ||u_j|| \}$
     - $\omega_- = \{i | i \in \omega_0, i \neq i_{max}\}$
     - $\omega_0 = \{i | i \in \omega_0, i \neq i_{max}\}$
     - Update $K_{\omega_0}^{-1} = \text{RankUpdate}(K_{\omega_0}^{-1}, V_{i_{max}})$

The above algorithm can compute a single vector, $\hat{b}_i$, which is just one column of $B$. To find the other columns, $RC_{\text{web}}$ applies the same approach to the modified (inflated) matrix, which for the $i^{th}$ column of $B$ is $V^{(i)} = V^{(i-1)} + VB_i\hat{b}_i^T$ for $i = 2, \ldots, P$. Thus, after the inference of each column of $B$ $RC_{\text{web}}$ modifies $V^{(i-1)}$ to $V^{(i)} (V^{(1)} \equiv V)$ so that the algorithm will not replicate its choice of $\omega_0$. Note that in the $i^{th}$ update of $V^{(i-1)}$ the $V_{\omega_0}^T$ will modify only the $\omega_-$ rows in $V^{(i-1)}$ since the rows of the $Vb_i\hat{b}_i^T$ whose indices are in $\omega_0$ contain only zero elements. Applying $RC_{\text{web}}$ to the inflated matrices avoids inferring...
multiple times the same \( \hat{b}_i \), but a \( \hat{b}_i \) inferred from the inflated matrix is generally going to differ at least slightly from the corresponding \( b_i \) that solves (4) for \( V \). To avoid that, \( RC_{web} \) uses the inflated matrices only for the first few iterations until the largest (by absolute magnitude) of the Pearson correlations between the smallest eigenvector of \( V_{\omega_0} \) from the current \((i^{th})\) iteration and the recovered columns of \( B \) is less than \( 1 - \epsilon \) and monotonically decreasing; \( \epsilon \) is chosen for numerical stability and also to reflect the similarity between the connectivity of \( R \) vertices that can be expected in the network whose topology is being recovered. A simpler alternative that works great in practice is to use the the inflated matrix for the first \( k \) iterations that are enough to find a new direction for \( \hat{b}_i \) and then \( RC_{web} \) switches back to \( V \) so that the solution is optimal for \( V \). The switch requires \( k \) rank update of \( K_{\omega_0}^{-1} \equiv (V_{\omega_0}^T V_{\omega_0})^{-1} \) and thus choosing \( k \) small saves computations. Choosing \( k \) too small, however, may not be enough to guarantee that \( \hat{b}_i \) will not recapitulate a solution that is already found. Usually \( k = 10 \) works great and can be easily increased if the new solution is very close to an old one.

There are a few notable elements that make \( RC_{web} \) efficient. First, \( RC_{web} \) does almost all computations in \( \mathbb{R}^P \) and since \( P < N \), \( P < M \), that saves both memory and CPU time. Second, each step requires only a few matrix-vector multiplication for computing the eigenvectors (since the change from the previous step is generally very small) and \( K_{\omega_0}^{-1} \) is computed by a rank–one update which obviates matrix inversion.

The approach that \( RC_{web} \) takes in solving (4) does not impose specific restrictions on the distribution of the observed variables (\( G \)), the noise in the data (\( Y \)) or the latent variables \( \hat{R} \). However, the initial approximation of \( b_i \) with \( v_i \) can be poor for data arising from dense networks or special worst-case datasets. As demonstrated theoretically (Benachy-Georges 2009) and tested numerically in the next section, \( RC_{web} \) performs very well at least in the absence of worst–case scenario special structures in the data.

### 4 VALIDATION

To evaluate the performance of \( RC_{web} \), I first apply it to data from simulated random bipartite networks with two different topology types, (1) Erdős & Rényi and (2) scale-free whose corresponding degree distributions are (1) Poisson and (2) power–law. The network topology is encoded in a weighted adjacency matrix \( C_{\text{gold}} \) and the values for the unobserved variables are drawn from a standard uniform distribution. The simulations result in data matrices \( G \in \mathbb{R}^{M \times N} \) containing \( M \) observations of all \( N \) unobserved variables. Accord-
\( \hat{C}_i \) is \( k = \arg \max_j (|\Sigma_{i1}|, \ldots, |\Sigma_{ij}|, \ldots, |\Sigma_{iP}|) \), where \( k \in \mathcal{R} \). The absolute value is required because the diagonal elements of \( \hat{D} \) can be negative. The accuracy is measured by the corresponding Pearson correlation, \( \rho_i = |\Sigma_{ik}| \). An optimal solution of this matching problem can be found by using belief propagation algorithm for the simple case of a bipartite graph even the LP relaxed version guarantees optimal solution (Sanghavi 2007). The overall accuracy is quantified by the mean correlation \( \bar{\rho} = (1/p) \sum_{i=1}^{p} \rho_i \) where \( p \) is the number of inferred unobserved variables and can equal to \( P \) or not depending on whether the number of unobserved variables is known or not. In computing \( \bar{\rho} \), each row of \( \mathcal{C}^{old} \) is allowed to correspond only to one row of \( \hat{C} \) and vice versa.

In addition to the two common limitations of permutation and scaling, some algorithms (d’Aspremont 2007) for sparse PCA require \( M > N \) and since this is not the case in many real world problems and in some of the datasets simulated here, those methods are not tested. Instead I chose emPCA, which does not require \( M > N \) and is the latest MLE algorithm for sparse PCA that according to its authors is more efficient than previous algorithms (Sigg 2008).

4.2 ACCURACY AND COMPLEXITY SCALING

\( \mathcal{RC}_{web} \) has a natural way for identifying the mean degree\(^3\). However, some of the other algorithms require the mean degree for optimal performance. To avoid underestimating an algorithm simply because it recovers networks that are too sparse or not sparse enough, I assume that the mean degree is known and it is input to all algorithms. First all algorithms are tested on a very easy inference problem, Fig.1. Since PSMF and BFRM have lower accuracy and PSMF is significantly slower than the other algorithms, the rest of the results will focus on the MLE algorithms that also have better performance. PSMF gives less accurate results with power–law networks which can be understood in terms of the uniform prior used by PSMF. PSMF has the advantage over the MLE algorithms in inferring a probabilistic network structure rather than a single estimate. Special advantage of BFRM is the seamless inclusion of response variables and measured factors in the inference. For the MLE algorithms, the accuracy of network inference increases with the ratio of observed to unobserved variables \( N/P \) (Fig.2) and with the number of observed configurations \( M \), Fig.3. In contrast, as the noise in the data and the mean out–degree (mean number of edges from \( \mathcal{R} \) to \( N \)) vertices) are increased, the accuracy of the inference decreases. All algorithms perform better on Poisson networks and the lower level of noise in the data from power–law networks was chosen to partially compensate for that. An important caveat when comparing the results for different algorithms is that K-SVD iteratively improves the accuracy of the solution, and thus the output is dependent on the maximum number of iterations allowed \( (I_{max}) \). For the results here, \( I_{max} = 20 \) and the accuracy of K-SVD may improve with higher number of iterations even though I did not observe significant improvement with \( I_{max} = 100 \). Even at 20 iterations K-SVD is significantly slower than \( \mathcal{RC}_{web} \) and emPCA, Fig.4. The scaling of the algorithms with respect to a parameter was determined by holding all other parameters constant and regressing the log of the CPU time against the log of the variable parameter, Fig.4. The scaling with respect to some parameters is below the theoretical expectation since the highest complexity steps may not be speed–limiting for the ranges of parameters used in the simulations.

4.3 INTERPRETABILITY

In analyzing real data the number of unobserved variables \( (P) \) may not be known. I observe that if a simulated network having \( P \) regulators is inferred assuming \( P^* > P \) regulators, the elements of \( \mathcal{B} \) corresponding to the excessive regulators are very close to zero, \( |B_{ij}| \leq 10^{-10} \) for \( i > P \) or \( j > P \). Thus, if the data truly originate from a sparse network \( \mathcal{RC}_{web} \) can discard unnecessary unobserved variables.

The results of \( \mathcal{RC}_{web} \) can be valuable even without

![Figure 1: Accuracy of network recovery as a function of the number of unobserved variables, \( P \). Number of observed variables \( N = 500 \); Number of observations, \( M = 280 \). All networks are with Poisson mean out–degree 0.10\(N = 50 \), with 10 % noise in the observations.](image-url)
identifying the physical factors corresponding to \( x_\mathcal{R} \). Yet identifying this correspondence, and thus overcoming the limitations outlined in section 4.1 can be very desirable. One practically relevant situation allowing in-depth interpretation of the results requires measuring (if only in a few configurations) the states of some of the variables that are generally unobserved \( (x_\mathcal{R}) \). This is relevant, for example, to situations in which measuring some variables is much more expensive than others (such as protein modifications versus messenger RNA concentrations) and some \( x_{s \in \mathcal{R}} \) can be observed only once or a few times. Assume that the states of the \( k^{th} \) physical factor are measured (data in vector \( u_k \)) in \( n_k \) number of configurations, whose indices are in the set \( \phi_k \). This information can be enough for determining the vertex \( x_{s \in \mathcal{R}} \) corresponding to the \( k^{th} \) factor and the corresponding \( \tilde{D}_{ss} \) as follows: 1) Compute the Pearson correlations \( \rho \) between \( u_k \) and the columns of \( \tilde{\mathbf{R}}_{\phi_k} \). Then, the vertex of the inferred network most likely to correspond to the \( k^{th} \) physical factor is \( s = \arg\max_i(|\rho_1|, \ldots, |\rho_i|, \ldots, |\rho_P|) \). 2)
common edges) of the regulator most likely to correspond to the \(k^{th}\) physical and its known connectivity (coming from prior knowledge) can be quantified by a p-val (the probability of observing such overlap by chance alone) computed from the hyper-geometric distribution. This approach is exemplified with gene-expression data in the next section.

4.4 INFERENCe OF A TRANSCRIPTIONAL NETWORK

Simulated models have the advantage of having known topology, and thus providing excellent basis for rigorous evaluation. Such rigorous evaluation is not possible for real biological networks. Yet, existing partial knowledge of transcriptional networks, can be combined with the approach outlined in section 4.3 for evaluation of inference of biological networks. In particular, consider the transcriptional network of budding yeast, Saccharomyces cerevisiae. It can be modeled as a two-layer network with transcription factors (TFs) and their complexes being regulators \((x_R)\) whose activities cannot be measured on a high-throughput scale because of technology limitations. TFs regulate the expression levels of messenger RNAs (mRNAs) whose concentrations can be measured easily on a high-throughput scale at multiple physiological states and represent the observed variables, \(x_N\). A partial knowledge about the connectivity in this transcriptional network comes from ChIP-chip experiments (MacIsaac 2006) which identify sets of genes regulated by the same transcription factors (TFs).

I first apply \(RC_{web}\) to a set mRNA configurations followed by the approach of section 4.3 to evaluate the results. The data matrix \(G\) contains 423 yeast datasets measured on the Affymetrix S98 microarray platform at a variety of physiological conditions. \(RC_{web}\) was initialized with \(P^* = 160\) unobserved variables (corresponding to TFs) and identified \(P = 153\) co-regulated sets of genes. The inferred adjacency matrix \(\hat{C}\) is compared directly to the adjacency matrix identified by ChIP-chip experiments (MacIsaac 2006) which identify sets of genes regulated by the same transcription factors (TFs).

The comparison between ChIP-chip gene sets and \(RC_{web}\) gene sets allows to identify the likely correspondence between \(x_R\) and TFs and to infer the activities of the TFs which are very hard to measure experimentally but crucially important to understanding the dynamics and logic of gene regulatory networks (Slavov 2009).

5 CONCLUSIONS

This paper introduces an approach \((RC_{web})\) for inferring latent (unobserved) factors explaining the behavior of observed variables. \(RC_{web}\) aims at inferring a sparse bipartite graph in which vertices connect inferred latent factors (e.g. regulators of mRNA transcription and degradation) to observed variables (e.g. target mRNAs). The salient difference distinguishing \(RC_{web}\) from prior related work is a new approach to attaining sparse solution that allows the natural inclusion of a generative model, relaxation of assumptions on distributions, and ultimately results in more accurate and computationally efficient inference compared to competing algorithms for sparse data decomposition.
Acknowledgements

I thank Dmitry Malioutov and Rodolfo Ros Zer-tuche for many insightful discussions and extensive, constructive feedback, as well as David M. Blei and Kenneth A. Dawson for support and comments. This work was supported by Irish Research Council for Science, Engineering and Technology (IRCSET) and by Science Foundation Ireland under Grant No.[SFI/SRC/B1155].

References

West M. (2002) Bayesian factor regression models in the large \( p \), small \( n \) paradigm. Bayesian Stat, 7, pp. 723–732

Carvalho C. et al. (2008) High-dimensional sparse factor modelling: Applications in gene expression genomics. JASA, 103, 484, pp. 1438–1456

Dueck D., Morris Q., Frey, B. (2005) Multi-way clustering of microarray data using probabilistic sparse matrix factorization. Bioinformatics, 21, pp. 144–151.

d’Aspremont A, Bach F., Ghaoui L. (2007) Optimal Solutions for Sparse Principal Component Analysis. JMLR 9, pp. 1269–1294

Sigg C., Buhmann J. (2008) Expectation-Maximization for Sparse and Non-Negative PCA. ICML, Helsinki, Finland

Srebro N. and Jaakkola T. (2001) Sparse Matrix Factorization for Analyzing Gene Expression Patterns, NIPS

Pe'er D., Regev A., and Tanay A. (2002) Minreg: Inferring an active regulator set. Bioinformatics, 18, 258–267

Bradley E., Trevor H., Iain J., Tibshirani R. (2004) Least Angle Regression. Annals of Statistics 32, pp. 407–499

Aharon M., Elad M., Bruckstein A. (2005) K-SVD and its non-negative variant for dictionary design. Wavelets XI 5914, pp. 591411–591424.

Banerjee, O., Ghaoui, L. E. & d’Aspremont, A. (2007) Model selection through sparse maximum likelihood estimation JMLR, 9, pp. 485–516

Cetin M., Malioutov D., Willsky A. (2002) A Variational Technique for Source Localization based on a Sparse Signal Reconstruction Perspective. ICASSP, 3, pp. 2965-2968, Orlando, Florida

Candès E., Wakin M. and Boyd S. (2007) Enhancing sparsity by reweighted \( \ell_1 \) minimization. J. Fourier Anal. Appl., 14, pp. 877–905.

Wilkinson, J. H. (1988) The Algebraic Eigenvalue Problem, Oxford University Press, USA, ISBN-13: 978-0198534181

Benaych-Georges F., Rao R. (2009) The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices, arXiv:0910.2120v1

Sanghavi S., Malioutov D., Willsky A. (2007) Linear programming analysis of loopy belief propagation for weighted matching, NIPS

MacIsaac K.D., et. al. (2006) An improved map of conserved regulatory sites for Saccharomyces cerevisiae. BMC Bioinformatics, 7:113. doi: 10.1186/1471-2105-7-113.

Slavov N., Dawson KA. (2009) Correlation Signature of the Macroscopic States of the Gene Regulatory Network in Cancer, PNAS, 106, 11, pp. 4079-4084