Learning Bi-clustered Vector Autoregressive Models

Tzu-Kuo Huang and Jeff Schneider

School of Computer Science, Carnegie Mellon University
{tzukuoh,schneide}@cs.cmu.edu

Abstract. Vector Auto-regressive (VAR) models are useful for analyzing temporal dependencies among multivariate time series, known as *Granger causality*. There exist methods for learning sparse VAR models, leading directly to causal networks among the variables of interest. Another useful type of analysis comes from clustering methods, which summarize multiple time series by putting them into groups. We develop a methodology that integrates both types of analyses, motivated by the intuition that Granger causal relations in real-world time series may exhibit some clustering structure, in which case the estimation of both should be carried out together. Our methodology combines sparse learning and a nonparametric *bi-clustered* prior over the VAR model, conducting full Bayesian inference via blocked Gibbs sampling. Experiments on simulated and real data demonstrate improvements in both model estimation and clustering quality over standard alternatives, and in particular biologically more meaningful clusters in a T-cell activation gene expression time series dataset than those by other methods.

Keywords: time-series analysis, vector auto-regressive models, bi-clustering, Bayesian non-parametrics, gene expression analysis

1 Introduction

Vector Auto-regressive (VAR) models are standard tools for analyzing multivariate time series data, especially their temporal dependencies, known as *Granger causality*[1] [7]. VAR models have been successfully applied in a number of domains, such as finance and economics [23,14], to capture and forecast dynamic properties of time series data. Recently, researchers in computational biology, using ideas from sparse linear regression, developed sparse estimation techniques for VAR models [5,11,22] to learn from high-dimensional genomic time series a small set of pairwise, directed interactions, referred to as gene regulatory networks, some of which lead to novel biological hypotheses.

While individual edges convey important information about interactions, it is often desirable to obtain an aggregate and more interpretable description of the network of interest. One useful set of tools for this purpose are graph clustering

[1] More precisely, *graphical* Granger causality for more than two time series.

methods [20], which identify groups of nodes or vertices that have similar types of connections, such as a common set of neighboring nodes in undirected graphs, and shared parent or child nodes in directed graphs. These methods have been applied in the analysis of various types of networks, such as [6], and play a key role in graph visualization tools [9].

Motivated by the wide applicability of the above two threads of work and the observation that their goals are tightly coupled, we develop a methodology that integrates both types of analyses, estimating the underlying Granger causal network and its clustering structure *simultaneously*. We consider the following first-order p-dimensional VAR model:

$$\mathbf{x}_{(t)} \;=\; \mathbf{x}_{(t-1)}A + \boldsymbol{\epsilon}_{(t)}, \quad \boldsymbol{\epsilon}_{(t)} \;\sim\; \mathcal{N}(\mathbf{0}, \sigma^2 I), \tag{1}$$

where $\mathbf{x}_{(t)} \in \mathbb{R}^{1\times p}$ denotes the vector of variables observed at time t, $A \in \mathbb{R}^{p\times p}$ is known as the transition matrix, whose non-zero entries encode Granger causal relations among the variables, and $\boldsymbol{\epsilon}_{(t)}$'s denote independent noise vectors drawn from a zero-mean Gaussian with a spherical covariance $\sigma^2 I$. Our goal is to obtain a transition matrix estimate $\widehat{A}$ that is both *sparse*, leading directly to a causal network, and *clustered* so that variables sharing a similar set of connections are grouped together. Since the rows and the columns of A indicate different roles of the variables, the former revealing how variables affect themselves and the latter showing how variables get affected, we consider the more general *bi-clustering* setting, which allows two different sets of clusters for rows and columns, respectively. We take a nonparametric Bayesian approach, placing over A a nonparametric bi-clustered prior and carrying out full posterior inferences via a blocked Gibbs sampling scheme. Our simulation study demonstrates that when the underlying VAR model exhibits a clear bi-clustering structure, our proposed method improves over some natural alternatives, such as adaptive sparse learning methods [24] followed by bi-clustering, in terms of model estimation accuracy, clustering quality, and forecasting capability. More encouragingly, on a real-world T-cell activation gene expression time series data set [18] our proposed method finds an interesting bi-clustering structure, which leads to a biologically more meaningful interpretation than those by some state-of-the art time series clustering methods.

Before introducing our method, we briefly discuss related work in Section 2. Then we define our bi-clustered prior in Section 3, followed by our sampling scheme for posterior inferences in Section 4. Lastly, we report our experimental results in Section 5 and conclude with Section 6.

2 Related work

There has been a lot of work on sparse estimation of causal networks under VAR models, and perhaps even more on graph clustering. However, to the best of our knowledge, none of them has considered the simultaneous learning scheme we propose here. Some of the more recent sparse VAR estimation work [11,22] takes into account dependency further back in time and can even select the right length

of history, known as the order of the VAR model. While developing our method around first-order VAR models, we observe that it can also learn higher-order bi-clustered models by, for example, assigning transition matrix entries across multiple time lags to the same bi-cluster.

Another large body of related work ([13,16,2], just to name a few) concerns bi-clustering (or co-clustering) a data matrix, which usually consists of relations between two sets of objects, such as user ratings on items, or word occurrences in documents. Most of this work models data matrix entries by mixtures of distributions with different *means*, representing, for example, different mean ratings by different user groups on item groups. In contrast, common regularization schemes or prior beliefs for VAR estimation usually assume zero-mean entries for the transition matrix, biasing the final estimate towards being stable. Following such a practice, our method models transition matrix entries as *scale mixtures* of zero-mean distributions.

Finally, clustering time series data has been an active research topic in a number of areas, in particular computational biology. However, unlike our Granger causality based bi-clustering method, most of the existing work, such as [17,3] and the references therein, focus on grouping together *similar* time series, with a wide range of similarity measures from simple linear correlation to complicated Gaussian process based likelihood scores. Differences between our method and existing similarity-based approaches are demonstrated in Section 5 through both simulations and experiments on real data.

3 Bi-clustered prior

We treat the transition matrix $A \in \mathcal{R}^{p \times p}$ as a random variable and place over it a "bi-clustered" prior, as defined by the following generative process:

$$\begin{aligned} \boldsymbol{\pi}_u &\sim \mathsf{Stick\text{-}Break}(\alpha_u), \qquad \boldsymbol{\pi}_v \sim \mathsf{Stick\text{-}Break}(\alpha_v), \\ \{u_i\}_{1 \le i \le p} &\overset{i.i.d}{\sim} \mathsf{Multinomial}(\boldsymbol{\pi}_u), \qquad \{v_j\}_{1 \le j \le p} \overset{i.i.d}{\sim} \mathsf{Multinomial}(\boldsymbol{\pi}_v), \\ \{\lambda_{kl}\}_{1 \le k,l \le \infty} &\overset{i.i.d.}{\sim} \mathsf{Gamma}(h, c), \qquad (2) \\ A_{ij} &\sim \mathsf{Laplace}(0, 1/\lambda_{u_i v_j}), \quad 1 \le i, j \le p. \qquad (3) \end{aligned}$$

The process starts by drawing row and column mixture proportions $\boldsymbol{\pi}_u$ and $\boldsymbol{\pi}_v$ from the "stick-breaking" distribution [21], denoted by $\mathsf{Stick\text{-}Break}(\alpha)$ and defined on an infinite-dimensional simplex as follows:

$$\begin{aligned} \beta_k &\sim \mathsf{Beta}(1, \alpha), \\ \pi_k &:= \beta_k \prod_{m<k} (1 - \beta_m), \quad 1 \le k \le \infty, \end{aligned} \qquad (4)$$

where $\alpha > 0$ controls the average length of pieces broken from the stick, and may take different values α_u and α_v for rows and columns, respectively. Such a prior allows for an infinite number of mixture components or clusters, and lets the data

Algorithm 1 Blocked Gibbs Sampler

Input: Data X and Y, hyper-parameters h, c, α_u, α_v, and initial values $A^{(0)}, L^{(0)}$, $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}, (\sigma^{(0)})^2$
Output: Samples from the full joint posterior $p(A, L, \mathbf{u}, \mathbf{v}, \sigma^2 \mid X, Y)$
Set iteration $t = 1$
repeat
  for $i = 1$ **to** p **do**
    $A_i^{(t)} \sim p(A_i \mid A_{1:(i-1)}^{(t)}, A_{(i+1):p}^{(t-1)}, \mathbf{u}^{(t-1)}, \mathbf{v}^{(t-1)}, (\sigma^{(t-1)})^2), L^{(t-1)}, X, Y)$
  end for
  for $i = 1$ **to** p **do**
    $u_i^{(t)} \sim p(u_i \mid A^{(t)}, \mathbf{u}_{1:(i-1)}^{(t)}, \mathbf{u}_{(i+1):p}^{(t-1)}, \mathbf{v}^{(t-1)}, (\sigma^{(t-1)})^2, L^{(t-1)}, X, Y)$
  end for
  for $j = 1$ **to** p **do**
    $v_j^{(t)} \sim p(v_j \mid A^{(t)}, \mathbf{u}^{(t)}, \mathbf{v}_{1:(j-1)}^{(t)}, \mathbf{v}_{(j+1):p}^{(t-1)}, (\sigma^{(t-1)})^2, L^{(t-1)}, X, Y)$
  end for
  $(\sigma^{(t)})^2 \sim p(\sigma^2 \mid A^{(t)}, \mathbf{u}^{(t)}, \mathbf{v}^{(t)}, L^{(t-1)}, X, Y)$
  $L^{(t)} \sim p(L \mid A^{(t)}, \mathbf{u}^{(t)}, \mathbf{v}^{(t)}, (\sigma^{(t)})^2, X, Y)$
  Increase iteration t
until convergence
Notations: superscript (t) denotes iteration, A_i denotes the i-th row of A, $A_{i:j}$ denotes the sub-matrix in A from the i-th until the j-th row, and $\mathbf{u}_{i:j}$ denotes $\{u_n\}_{i \le n \le j}$.

decide the number of *effective* components having positive probability masses, thereby increasing modeling flexibility. The process then samples row-cluster and column-cluster indicator variables u_i's and v_j's from mixture proportions $\boldsymbol{\pi}_u$ and $\boldsymbol{\pi}_v$, and for the k-th row-cluster and the l-th column-cluster draws an inverse-scale, or rate parameter λ_{kl} from a Gamma distribution with shape parameter h and scale parameter c. Finally, the generative process draws each matrix entry A_{ij} from a zero-mean Laplace distribution with inverse scale $\lambda_{u_i v_j}$, such that entries belonging to the same bi-cluster share the same inverse scale, and hence represent interactions of similar *magnitudes*, whether positive or negative.

The above bi-clustered prior subsumes a few interesting special cases. In some applications researchers may believe the clusters should be symmetric about rows and columns, which corresponds to enforcing $\mathbf{u} = \mathbf{v}$. If they further believe that within-cluster interactions should be stronger than between-cluster ones, they may adjust accordingly the hyper-parameters in the Gamma prior (2), or as in the group sparse prior proposed by [12] for Gaussian precision estimation, simply require all within-cluster matrix entries to have the same inverse scale constrained to be smaller than the one shared by all between-cluster entries. Our inference scheme detailed in the next section can be easily adapted to all these special cases.

There can be interesting generalizations as well. For example, depending on the application of interest, it may be desirable to distinguish positive interactions from negative ones, so that a bi-cluster of transition matrix entries possess not only similar strengths, but also *consistent signs*. However, such a generalization

requires a more delicate per-entry prior and therefore a more complex sampling scheme, which we leave as an interesting direction for future work.

4 Posterior inference

Let L denote the collection of λ_{kl}'s, $\mathbf{u}$ and $\mathbf{v}$ denote $\{u_i\}_{1\le i\le p}$ and $\{v_j\}_{1\le j\le p}$, respectively. Given one or more time series, collectively denoted as matrices X and Y whose rows represent successive pairs of observations, i.e.,

$$Y_i \;=\; X_i A + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \;\sim\; \mathcal{N}(\mathbf{0}, \sigma^2 I),$$

we aim to carry out posterior inferences about the transition matrix A, and row and column cluster indicators $\mathbf{u}$ and $\mathbf{v}$. To do so, we consider sampling from the full joint posterior $p(A, L, \mathbf{u}, \mathbf{v}, \sigma^2 \mid X, Y)$, and develop an efficient blocked Gibbs sampler outlined in Algorithm 1. Starting with some reasonable initial configuration, the algorithm iteratively samples rows of A, row and column-cluster indicator variables $\mathbf{u}$ and $\mathbf{v}$, the noise variance[2] σ^2, and the inverse scale parameters L from their respective conditional distributions. Next we describe in more details sampling from those conditional distributions.

4.1 Sampling the transition matrix A

Let A_{-i} denote the sub-matrix of A excluding the i-th row, X'_i and X'_{-i} denote the i-th column of X and the sub-matrix of X excluding the i-th column. Algorithm 1 requires sampling from the following conditional distribution:

$$p(A_i \mid A_{-i}, \mathbf{u}, \mathbf{v}, \sigma^2, L, X, Y) \;\propto\; \prod_{1\le j\le p} \mathcal{N}(A_{ij} \mid \mu_{ij}, \sigma_i^2)\mathsf{Laplace}(A_{ij} \mid 0, 1/\lambda_{u_i v_j}),$$

where

$$\mu_{ij} \;:=\; (X'_i/\|X'_i\|_2^2)^\top (Y - X'_{-i}A_{-i})'_j, \quad \sigma_i^2 \;:=\; \sigma^2/\|X'_i\|^2.$$

Therefore, all we need is sampling from univariate densities of the form:

$$f(x) \;\propto\; \mathcal{N}(x \mid \mu, \sigma^2)\mathsf{Laplace}(x \mid 0, 1/\lambda), \tag{5}$$

whose c.d.f. $F(x)$ can be expressed in terms of the standard normal c.d.f. $\Phi(\cdot)$:

$$F(x) \;=\; \frac{C_1}{C}\Phi\Big(\frac{x^- - (\mu + \sigma^2\lambda)}{\sigma}\Big) + \frac{C_2}{C}\Big(\Phi\Big(\frac{x^+ - (\mu - \sigma^2\lambda)}{\sigma}\Big) - \Phi\Big(-\frac{\mu - \sigma^2\lambda}{\sigma}\Big)\Big),$$

where $x^- := \min(x, 0)$, $x^+ := \max(x, 0)$, and

$$C := C_1\Phi\Big(-\frac{\mu + \sigma^2\lambda}{\sigma}\Big) + C_2\Big(1 - \Phi\Big(-\frac{\mu - \sigma^2\lambda}{\sigma}\Big)\Big),$$
$$C_1 := \frac{\lambda}{2}\exp\Big(\frac{\lambda(2\mu + \sigma^2\lambda)}{2}\Big), \quad C_2 := \frac{\lambda}{2}\exp\Big(\frac{\lambda(\sigma^2\lambda - 2\mu)}{2}\Big).$$

[2] Our sampling scheme can be easily modified to handle diagonal covariances.

We then sample from $f(x)$ with the inverse c.d.f. method. To reduce the potential sampling bias introduced by a fixed sampling schedule, we follow a random ordering of the rows of A in each iteration.

Algorithm 1 generates samples from the full joint posterior, but sometimes it is desirable to obtain a point estimate of A. One simple estimate is the (empirical) posterior mean; however, it is rarely sparse. To get a sparse estimate, we carry out the following "sample EM" step after Algorithm 1 converges:

$$\widehat{A}^{\text{Biclus-EM}} \;:=\; \arg\max_{A} \sum_{t} \log p(A \mid \mathbf{u}^{(t)}, \mathbf{v}^{(t)}, (\sigma^{(t)})^2, L^{(t)}, X, Y), \qquad (6)$$

where t starts at a large number and skips some fixed number of iterations to give better-mixed and more independent samples. The optimization problem (6) is in the form of sparse least square regression, which we solve with a simple coordinate descent algorithm.

4.2 Sampling row and cluster indicators

Since our sampling procedures for $\mathbf{u}$ and $\mathbf{v}$ are symmetric, we only describe the one for $\mathbf{u}$. It can be viewed as an instantiation of the general Gibbs sampling scheme in [13]. According to our model assumption, $\mathbf{u}$ is independent of the data X, Y and the noise variance σ^2 conditioned on all other random variables. Moreover, under the stick-breaking prior (4) over the row mixture proportions $\boldsymbol{\pi}_u$ and some fixed $\mathbf{v}$, we can view $\mathbf{u}$ and the rows of A as cluster indicators and samples drawn from a Dirichlet process mixture model with $\mathsf{Gamma}(h, c)$ as the base distribution over cluster parameters. Finally, the Laplace distribution and the Gamma distribution are conjugate pairs, allowing us to integrate out the inverse scale parameters L and derive the following "collapsed" sampling scheme:

$$p(u_i = k' \in \text{existing row-clusters} \mid A, \mathbf{u}_{-i}, \mathbf{v})$$
$$\propto \left(\prod_{k,l} \frac{\Gamma((N_{-i}[k] + \delta_{kk'})m_l + h)/(\Gamma(h)c^h)}{\Big(\|A_{-i}[k,l]\|_1 + \delta_{kk'}\|A_i[l]\|_1 + 1/c\Big)^{(N_{-i}[k]+\delta_{kk'})M[l]+h}} \right) \frac{N_{-i}[k']}{p - 1 + \alpha_u},$$

$$p(u_i = \text{a new row-cluster} \mid A, \mathbf{u}_{-i}, \mathbf{v})$$
$$\propto \left(\prod_{k,l} \frac{\Gamma(N_{-i}[k]M[l] + h)/(\Gamma(h)c^h)}{\Big(\|A_{-i}[k,l]\|_1 + 1/c\Big)^{N_{-i}[k]M[l]+h}} \cdot \frac{\Gamma(M[l] + h)/(\Gamma(h)c^h)}{\Big(\|A_i[l]\|_1 + 1/c\Big)^{M[l]+h}} \right) \frac{\alpha_u}{p - 1 + \alpha_u},$$

where $\Gamma(\cdot)$ is the Gamma function, δ_{ab} denotes the Kronecker delta function, $N_{-i}[k]$ is the size of the k-th row-cluster excluding A_i, $M[l]$ is the size of the l-th column-cluster, and

$$\|A_{-i}[k,l]\|_1 \;:=\; \sum_{s \neq i, u_s = k, v_j = l} |A_{sj}|, \qquad \|A_i[l]\|_1 \;:=\; \sum_{v_j = l} |A_{ij}|.$$

As in the previous section, we randomly permute u_i's and v_j's in each iteration to reduce sampling bias, and also randomly choose to sample $\mathbf{u}$ or $\mathbf{v}$ first.

Just as with the transition matrix A, we may want to obtain point estimates of the cluster indicators. The usual empirical mean estimator does not work here because the cluster labels may change over iterations. We thus employ the following procedure:

1. Construct a similarity matrix S such that

$$S_{ij} \;:=\; \frac{1}{T}\sum_{t} \delta_{u_i^{(t)} v_j^{(t)}}, \qquad 1 \le i, j, \le p,$$

   where t selects iterations to approach mixing and independence as in (6), and T is the total number of iterations selected.
2. Run normalized spectral clustering [15] on S, with the number of clusters set according to the spectral gap of S.

4.3 Sampling noise variance and inverse scale parameters

On the noise variance σ^2 we place an inverse-Gamma prior with shape $a > 0$ and scale $\beta > 0$, leading to the following posterior:

$$\sigma^2 \mid A, X, Y \;\sim\; \mathsf{I\text{-}Gamma}(a + pT/2, 2\|Y - XA\|_F^{-2} + \beta), \tag{7}$$

where T is the number of rows in X and $\|\cdot\|_F$ denotes the matrix Frobenius norm. Due to the conjugacy mentioned in the last section, the inverse scale parameters λ_{kl}'s have the following posterior:

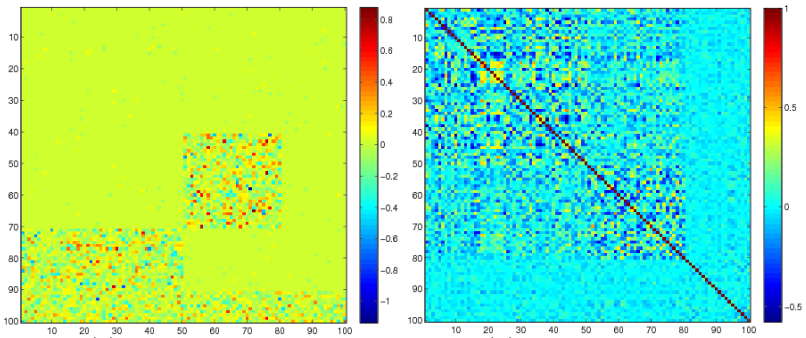
$$\lambda_{kl} \mid A, \mathbf{u}, \mathbf{v} \;\sim\; \mathsf{Gamma}(N[k]M[l] + h, (\|A[k,l]\|_1 + 1/c)^{-1}).$$

5 Experiments

We conduct both simulations and experiments on a real gene expression time series dataset, and compare the proposed method with two types of approaches:
Learning VAR by sparse linear regression, followed by bi-clustering
Unlike the proposed method, which makes inferences about the transition matrix A and cluster indicators jointly, this natural baseline method first estimates the transition matrix by adaptive sparse or L_1 linear regression [24]:

$$\widehat{A}^{L_1} \;:=\; \arg\min_A \frac{1}{2}\|Y - XA\|_F^2 + \lambda \sum_{i,j} \frac{|A_{ij}|}{|\widehat{A}^{\text{ols}}_{ij}|^\gamma}, \tag{8}$$

where $\widehat{A}^{\text{ols}}$ denotes the ordinary least-square estimator, and then bi-clusters $\widehat{A}^{L_1}$ by either the cluster indicator sampling procedure in Section 4.2 or standard clustering methods applied to rows and columns separately. We compare the proposed method and this baseline in terms of predictive capability, clustering



(a) Transition matrix (b) Correlation matrix

Fig. 1. Heat maps of the synthetic bi-clustered VAR

performance, and in the case of simulation study, model estimation error.

Clustering based on time series similarity
As described in Section 2, existing time series clustering methods are designed to group together time series that exhibit a similar behavior or dependency over time, whereas our proposed method clusters time series based on their (Granger) causal relations. We compare the proposed method with the time series clustering method proposed by [3], which models time series data by Gaussian processes and performs Bayesian Hierarchical Clustering [8], achieving state-of-the art clustering performances on the real genes time series data used in Section 5.

5.1 Simulation

We generate a transition matrix A of size 100 by first sampling entries in bi-clusters:

$$A_{ij} \sim \begin{cases} \mathsf{Laplace}(0, \sqrt{60}^{-1} i), & 41 \le i \le 70, 51 \le j \le 80, \\ \mathsf{Laplace}(0, \sqrt{70}^{-1}), & 71 \le i \le 90, 1 \le j \le 50, \\ \mathsf{Laplace}(0, \sqrt{110}^{-1}), & 91 \le i \le 100, 1 \le j \le 100, \end{cases} \tag{9}$$

and then all the remaining entries from a sparse back-ground matrix:

$$A_{ij} = \begin{cases} B_{ij} & \text{if } |B_{ij}| \ge q_{98}\big(\{|B_{i'j'}|\}_{1 \le i', j' \le 100}\big), \\ 0 & \text{otherwise,} \end{cases} \qquad i, j \text{ not covered in (9)},$$

where

$$\{B_{ij}\}_{1 \le i,j, \le 100} \overset{i.i.d.}{\sim} \mathsf{Laplace}(0, (5\sqrt{200})^{-1})$$

and $q_{98}(\cdot)$ denotes the 98-th percentile. Figure 1(a) shows the heat map of the actual A we obtain by the above sampling scheme, showing clearly four row-clusters and three column-clusters. This transition matrix has the largest eigenvalue modulus of 0.9280, constituting a stable VAR model.

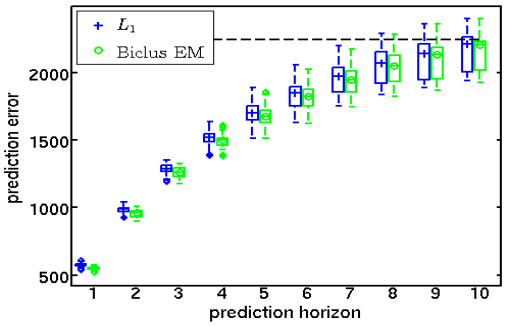


Fig. 2. Prediction errors up to 10 time steps. Errors for longer horizons are close to those by the mean (zero) prediction, shown in black dashed line, and are not reported.

We then sample 10 independent time series of 50 time steps from the VAR model (1), with noise variance $\sigma^2 = 5$. We initialize each time series with an independent sample drawn from the stationary distribution of (1), whose correlation matrix is shown in Figure 1(b), suggesting that clustering based on correlations among time series may not recover the bi-cluster structure in Figure 1(a).

To compare the proposed method with the two baselines described in the beginning of Section 5, we repeat the following experiment 20 times: a random subset of two time series are treated as testing data, while the other eight time series are used as training data. For L_1 linear regression (8) we randomly hold out two time series from the training data as a validation set for choosing the best regularization parameter λ from $\{2^{-2}, 2^{-1}, \ldots, 2^{10}\}$ and weight-adaption parameter γ from $\{0, 2^{-2}, 2^{-1}, \ldots, 2^{2}\}$, with which the final $\widehat{A}^{L_1}$ is estimated from all the training data. To bi-cluster $\widehat{A}^{L_1}$, we consider the following:

- $\mathbf{L_1}$**+Biclus**: run the sampling procedure in Section 4.2 on $\widehat{A}^{L_1}$.
- **Refit+Biclus**: refit the non-zero entries of $\widehat{A}^{L_1}$ using least-square, and run the sampling procedure in Section 4.2.
- $\mathbf{L_1}$ **row-clus** (**col-clus**): construct similarity matrices

$$S^u_{ij} := \sum_{1 \le s \le p} |\widehat{A}^{L_1}_{is}||\widehat{A}^{L_1}_{js}|, \quad S^v_{ij} := \sum_{1 \le s \le p} |\widehat{A}^{L_1}_{si}||\widehat{A}^{L_1}_{sj}|, \quad 1 \le i, j \le p.$$

  Then run normalized spectral clustering [15] on S^u and S^v, with the number of clusters set to 4 for rows and 3 for columns, respectively.

For the second baseline, Bayesian Hierarchical Clustering and Gaussian processes (GPs), we use the R package BHC (version 1.8.0) with the squared-exponential covariance for Gaussian processes, as suggested by the author of the package. Following [3] we normalize each time series to have mean 0 and standard deviation 1. The package can be configured to use replicate information (multiple series) or not, and we experiment with both settings, abbreviated as BHC-SE reps and BHC-SE, respectively. In both settings we give the BHC package the

Table 1. Model estimation error on simulated data

	Normalized matrix error	Signed-support error
L_1	0.3133±0.0003	0.3012±0.0008
Biclus EM	**0.2419±0.0003**	**0.0662±0.0012**

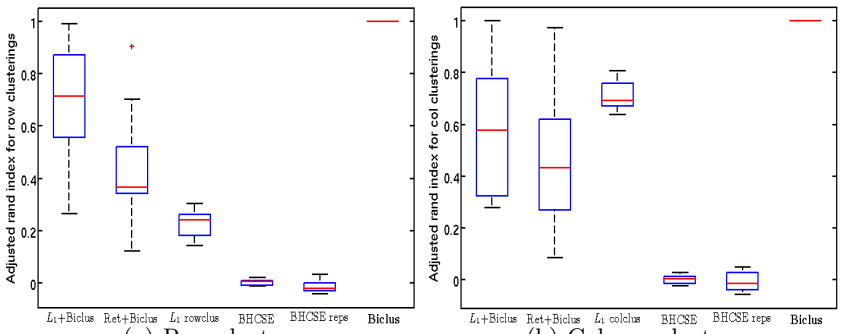
mean of the eight training series as input, but additionally supply BHC-SE reps a noise variance estimated from multiple training series to aid GP modeling.

In our proposed method, several hyper-parameters need to be specified. For the stick-breaking parameters α_u and α_v, we find that values in a reasonable range often lead to similar posterior inferences, and simply set both to be 1.5. We set the noise variance prior parameters in (7) to be $a = 9$ and $\beta = 10$. For the two parameters in the Gamma prior (2), we set $h = 2$ and $c = \sqrt{2p} = \sqrt{200}$ to bias the transition matrices sampled from the Laplace prior (3) towards being stable. Another set of inputs to Algorithm 1 are the initial values, which we set as follows: $A^{(0)} = \mathbf{0}$, $\mathbf{u}^{(0)} = \mathbf{v}^{(0)} = \mathbf{1}$, $(\sigma^{(0)})^2 = 1$, and $L^{(0)} = (h-1)c = \sqrt{200}$. We run Algorithm 1 and the sampling procedures for L_1+Biclus and Refit+Biclus for 2,500 iterations, and take samples in every 10 iterations starting from the 1,501-st iteration, at which the sampling algorithms have mixed quite well, to compute point estimates for A, $\mathbf{u}$ and $\mathbf{v}$ as described in Sections 4.1 and 4.2.

Figure 2 shows the squared prediction errors of L_1 linear regression (L_1) and the proposed method with a final sample EM step (Biclus EM) for various prediction horizons up to 10. Predictions errors for longer horizons are close to those by predicting the mean of the series, which is zero under our stable VAR model, and are not reported here. Biclus EM slightly outperforms L_1, and paired t tests show that the improvements for all 10 horizons are significant at a p-value ≤ 0.01. This suggests that when the underlying VAR model does have a bi-clustering structure, our proposed method can improve the prediction performance over adaptive L_1 regression, though by a small margin.

Another way to compare L_1 and Biclus EM is through model estimation error, and we report in Table 1 these two types of error:
Normalized matrix error: $\|\widehat{A} - A\|_F / \|A\|_F$,
Signed-support error: $\frac{1}{p^2} \sum_{1 \leq i,j \leq p} \mathbb{I}(\mathsf{sign}(\widehat{A}_{ij}) \neq \mathsf{sign}(A_{ij}))$.
Clearly, Biclus EM performs much better than L_1 in recovering the underlying model, and in particular achieves a huge gain in signed support error, thanks to its use of bi-clustered inverse scale parameters L.

Perhaps the most interesting is the clustering quality, which we evaluate by the *Adjusted Rand Index* [10], a common measure of similarity between two clusterings based on co-occurrences of object pairs across clusterings, with correction for chance effects. An adjusted Rand index takes the maximum value of 1 only when the two clusterings are identical (modulo label permutation), and is close to 0 when the agreement between the two clusterings could have resulted from two random clusterings. Figure 3 shows the clustering performances of different methods. The proposed method, labeled as Biclus, outperforms all alternatives greatly and always recovers the correct row and column clusterings. The two-



(a) Row clusters (b) Column clusters

Fig. 3. Adjusted Rand index on simulated data

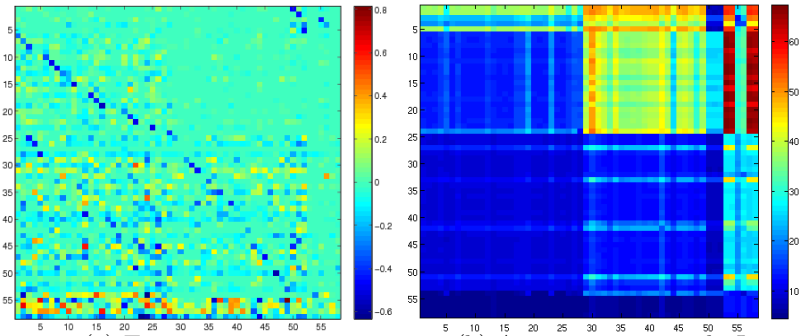
stage baseline methods L_1+Biclus, Refit+Biclus, and L_1 row-clus (col-clus) make a significant amount of errors, but still recover moderately accurate clusterings. In contrast, the clusterings by the time-series similarity based methods, BHC-SE and BHC-SE reps, are barely better than random clusterings. To explain this, we first point out that BHC-SE and BHC-SE reps are designed to model time series as noisy observations of deterministic, time-dependent "trends" or "curves" and to group similar curves together, but the time series generated from our stable VAR model all have zero expectation *at all time points* (not just *across time*). As a result, clustering based on similar trends may just be fitting noise in our simulated series. These results on clustering quality suggest that when the underlying cluster structure stems from (Granger) causal relations, clustering methods based on series similarity may give irrelevant results, and we really need methods that explicitly take into account dynamic interaction patterns, such as the one we propose here.

5.2 Modeling T-cell activation gene expression time series

We analyze a gene expression time series dataset[3] collected by [18] from a T-cell activation experiment. To facilitate the analysis, they pre-processed the raw data to obtain 44 replicates of 58 gene time series across 10 unevenly-spaced time points. Recently [3] carried out clustering analysis of these time series data, with their proposed Gaussian process (GP) based Bayesian Hierarchical Clustering (BHC) and quite a few other state-of-the art time series clustering methods. BHC, aided by GP with a cubic spline covariance function, gave the best clustering result as measured by the Biological Homogeneity Index (BHI) [4], which scores a gene cluster based on its number of gene pairs that share certain biological annotations (Gene Ontology terms).

To apply our proposed method, we first normalize each time series to have mean 0 and standard deviation 1 across both time points and replicates, and

[3] Available in the R package longitudinal.



(a) Transition matrix (b) Average inverse scale L

Fig. 4. Heat maps of the Biclus-EM estimate of A and the inverse scale parameters L averaged over posterior samples; rows and columns permuted according to clusters.

then "de-trend" the series by taking the first order difference, resulting in 44 replicates of 58 time series of gene expression differences across 9 time points. We run Algorithm 1 on this de-trended dataset, with all the hyper-parameters and initial values set in the same way as in our simulation study. In 3,000 iterations the algorithm mixes reasonably well; we let it run for another 2,000 iterations and take samples from every 10 iterations, resulting in 200 posterior samples, to compute point estimates for A, cluster indicators $\mathbf{u}$ and $\mathbf{v}$ as described in Sections 4.1 and 4.2. Figures 4(a) and 4(b) show the heat maps of the transition matrix point estimate and the inverse scale parameters λ_{ij}'s averaged over the posterior samples, with rows and columns permuted according to clusters, revealing a quite clear bi-clustering structure.

For competing methods, we use the GP based Bayesian Hierarchical Clustering (BHC) by [3], with two GP covariance functions: cubic spline (BHC-C) and squared-exponential (BHC-SE)[4]. We also apply the two-stage method L_1+Biclus described in our simulation study, but its posterior samples give an average of 15 clusters, which is much more than the number of clusters, around 4, from the spectral analysis described in Section 4.2, suggesting a high level of uncertainty in their posterior inferences about cluster indicators. We thus do not report their results here. The other two simple baselines are: Corr, standing for normalized spectral clustering on the correlation matrix of the 58 time series averaged over all 44 replicates, the number of clusters 2 determined by the spectral gap, and All-in-one, which simply puts all genes in one cluster.

Figure 5 shows the BHI scores[5] given by different methods, and higher-values indicate bettering clusterings. Biclus row and Biclus col respectively denote the

[4] Here we only report results obtained without using replicate information because using replicate information does not give better results. We obtain cluster labels from `http://www.biomedcentral.com/1471-2105/12/399/additional`.

[5] We compute BHIs by the BHI function in the R package clValid (version 0.6-4) [1] and the database hgu133plus2.db (version 2.6.3), following [3].

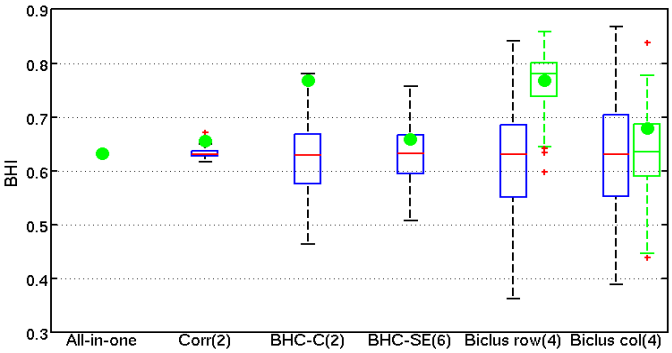


Fig. 5. BHI. Green dots show BHIs of different methods; blue boxes are BHIs obtained by 200 random permutations of cluster labels by those methods; green boxes are BHIs computed on posterior cluster indicator samples from the proposed method. In parentheses are numbers of clusters given by different methods.

row and column clusterings given by our method. To measure the significance of the clusterings, we report BHI scores computed on 200 random permutations of the cluster labels given by each method. For Biclus row and Biclus col, we also report the scores computed on the 200 posterior samples. All-in-one has a BHI score around 0.63, suggesting that nearly two-thirds of all gene pairs share some biological annotations. Corr puts genes into two nearly equal-sized clusters (28 and 30), but does not increase the BHI score much. In contrast, BHC-C and Biclus row achieve substantially higher scores, and both are significantly better than those by random permutations, showing that the improvements are much more likely due to the methods rather than varying numbers or sizes of clusters. We also note that even though Corr and BHC-C both give two clusters, the two BHC-C clusters have very different sizes (48 and 10), which cause a larger variance in their BHI distribution under random label permutations. Lastly, BHC-SE and Biclus col give lower scores that are not significantly better than random permutations. One possible explanation for the difference in scores by Biclus row and Biclus col is that the former bases itself on how genes *affect* one another while the latter on how genes *are affected* by others, and Gene Ontology terms, the biological annotations underlying the BHI function, describe more about genes' active roles or molecular functions in various biological processes than what influence genes.

Finally, to gain more understanding on the clusters by BHC-C and Biclus row, we conduct gene function profiling with the web-based tool **g:Profiler** [19], which performs "statistical enrichment analysis to provide interpretation to user-defined gene lists." We select the following three options: *Significant only*, *Hierarchical sorting*, and *No electronic GO annotations*. For BHC-C, 4 out of 10 genes in the small cluster are found to be associated with the KEGG cell-cycle pathway (04110), but the other 6 genes are not mapped to collectively meaningful annotations. The profiling results of the large BHC-C cluster with 48 genes are in

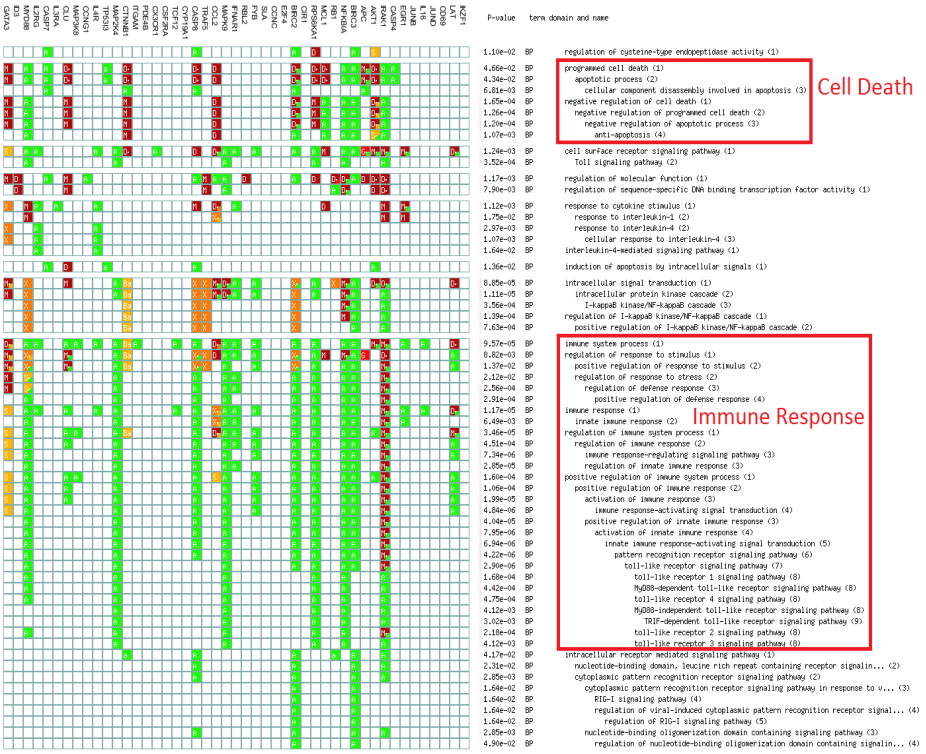
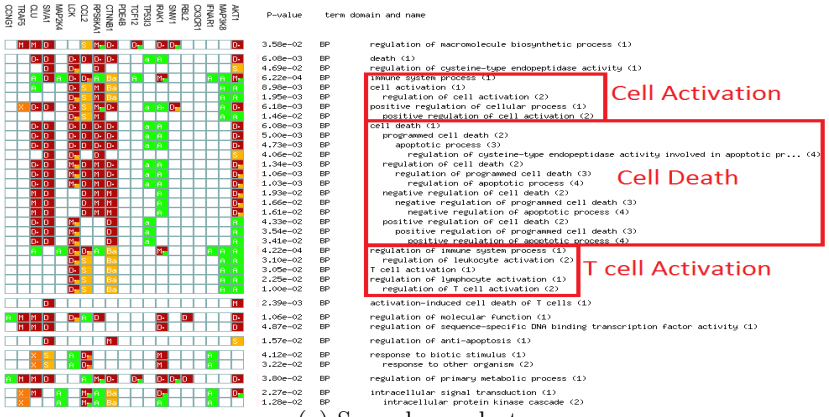


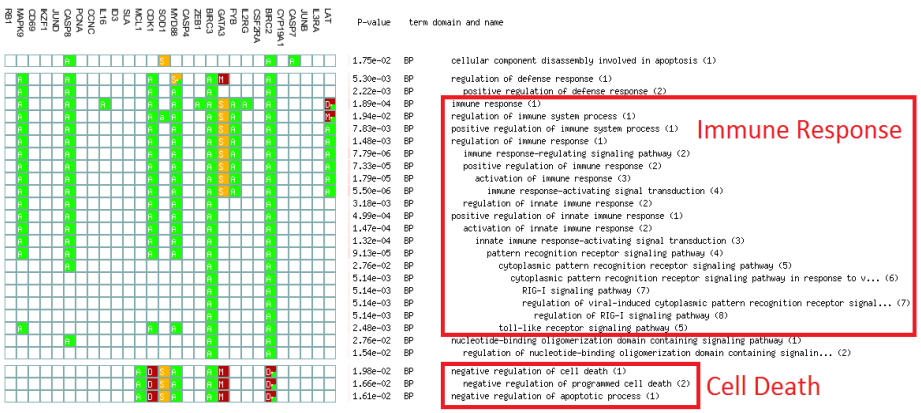
Fig. 6. Gene functional profiling of the large BHC-C cluster

Figure 6; for better visibility we show only the Gene Ontology (GO) terms and high-light similar terms with red rectangles and tags. About a half of the terms are related to cell death and immune response, and the other half are lower-level descriptions involving, for example, signaling pathways. For Biclus row, we report the profiling results of only the two larger clusters (the second and the third) in Figure 7, because the two smaller clusters, each containing 5 genes, are not mapped to collectively meaningful GO terms. Interestingly, the two large Biclus row clusters are associated with T-cell activation and immune response respectively, and together they cover 41 of the 48 genes in the large BHC-C cluster. This suggests that our method roughly splits the large BHC-C cluster into two smaller ones, each being mapped to a more focused set of biological annotations. Moreover, these Biclus profiling results, the heat map in Figure 4(a), and the contingency table (shown in the right) between the row and column clusters altogether constitute a nice resonance with the fact that T-cell activation results from, rather than leading to, the emergence of immune responses.

row \ col	1	2	3	4
1	0	0	3	2
2	17	2	0	0
3	10	17	0	2
4	1	2	0	2



(a) Second row cluster



(b) Third row cluster

Fig. 7. Gene functional profiling of two large row clusters by the proposed method

6 Conclusion

We develop a nonparametric Bayesian method to simultaneously infer sparse VAR models and bi-clusterings from multivariate time series data, and demonstrate its effectiveness via simulations and experiments on real T-cell activation gene expression time series, on which the proposed method finds a more biologically interpretable clustering than those by some state-of-the art methods. Future directions include modeling signs of transition matrix entries, generalizations to higher-order VAR models, and applications to other real time series.

References

1. G. Brock, V. Pihur, S. Datta, and S. Datta. clvalid: An R package for cluster validation. *Journal of Statistical Software*, 25(4):1–22, 2008.

2. S. Busygin, O. Prokopyev, and P. Pardalos. Biclustering in data mining. *Computers & Operations Research*, 35(9):2964–2987, 2008.
3. E. Cooke, R. Savage, P. Kirk, R. Darkins, and D. Wild. Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. *BMC bioinformatics*, 12(1):399, 2011.
4. S. Datta and S. Datta. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *BMC bioinformatics*, 7(1):397, 2006.
5. A. Fujita, J. Sato, H. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. Sogayar, and C. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*, 1(1):39, 2007.
6. M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821, 2002.
7. C. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, pages 424–438, 1969.
8. K. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *The 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.
9. I. Herman, G. Melançon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.
10. L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
11. A. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110, 2009.
12. B. M. Marlin, M. Schmidt, and K. P. Murphy. Group sparse priors for covariance estimation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, Montreal, Canada, 2009.
13. E. Meeds and S. Roweis. Nonparametric Bayesian biclustering. Technical report, Department of Computer Science, University of Toronto, 2007.
14. T. C. Mills. *The Econometric Modelling of Financial Time Series*. Cambridge University Press, second edition, 1999.
15. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 2001.
16. I. Porteous, E. Bart, and M. Welling. Multi-hdp: A non-parametric bayesian model for tensor factorization. In *Proc. of the 23rd National Conf. on Artificial Intelligence*, pages 1487–1490, 2008.
17. M. Ramoni, P. Sebastiani, and I. Kohane. Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Sciences*, 99(14):9121, 2002.
18. C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. Wild, and F. Falciani. Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361–1372, 2004.
19. J. Reimand, T. Arak, and J. Vilo. g: Profiler – a web server for functional interpretation of gene lists (2011 update). *Nucleic acids research*, 39(suppl 2):W307–W315, 2011.
20. S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.
21. J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.
22. A. Shojaie, S. Basu, and G. Michailidis. Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. *Statistics in Biosciences*, pages 1–18, 2011.
23. R. S. Tsay. *Analysis of financial time series*. Wiley-Interscience, 2005.
24. H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.