Sparse VAR models and variants

- Lasso estimation in i.i.d. setting. Basic lasso, adaptive lasso, debiased lasso etc.
- Extension to high-dimensional VAR model.
- Some examples and extensions

Basics of Lasso - Framework

- Consider regression problem with sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ data with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$: p-dimensional predictors and response vector $\mathbf{y} = (y_1, \ldots, y_N)'$.
- OLS estimator is given by

$$\arg\min \frac{1}{2N} \|\mathbf{y} - \beta_0 \mathbf{1} - X\boldsymbol{\beta}\|_2^2,$$

  where $X = (\mathbf{x}_1' \cdots \mathbf{x}_N')'$ be $N \times p$ design matrix and $\mathbf{1} = (1, \ldots, 1)'$ be vectoor of ones.
- Lasso estimator is defined as

$$\boxed{\begin{aligned} &\text{minimize} \frac{1}{2N} \|\mathbf{y} - \beta_0 \mathbf{1} - X\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \\ &L_1\text{-constraint} \quad \|\boldsymbol{\beta}_1\|_1 = \sum_{j=1}^{p} |\beta_j| \le t \end{aligned} \qquad (1)}$$

Lasso -Lagrangian form

- For the notational convenience, we standardize predictor X (centered and unit variance) and $\mathbf{y}$ is also cnetered $(\mathbf{y} - \overline{\mathbf{y}})$. Then, it gives $\widehat{\beta} = 0$.
- Then, we can rewrite Lasso problem into Lagrange form

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\text{minimize}} \frac{1}{2N} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \quad \lambda \geq 0 \tag{2}$$

- See Chapter 2 of Hastie et al. (2015) for details. Few Remarks in order.

Lasso: remarks

1. Lagrange duality means 1-1 correspondence between (1) and (2). For each t such that $\|\boldsymbol{\beta}\|_1 \leq t$, we can find corresponding λ in (2) . If $\widehat{\boldsymbol{\beta}}_\lambda$ solves (2) for given λ, then it solves (1) with $t = \|\hat{\boldsymbol{\beta}}_\lambda\|_1$
2. The equivalence is non-trivial but well-known. See section 5.5.3 in Boyd et al. (2004)
3. $\frac{1}{2N}$ is used to cancel out derivative.
4. Hence, we can solve lasso by finding the solution of (2). The "derivative $=0$" gives the solution because

$$\frac{1}{2N}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

is convex function. Why?

Lasso: remarks

Hence, only need to solve

$$\begin{cases} -\frac{1}{N}\langle \mathbf{x}_j, \mathbf{y} - X\boldsymbol{\beta}\rangle + \lambda \mathrm{sign}(\boldsymbol{\beta}_j) = 0, j = 1, \ldots, p & \text{if } \beta_j \neq 0, \\ \text{any value in } [-1, +1] & \text{if } \beta_j = 0, \end{cases}$$

where $[-1, +1]$ is the subgradient of absolute function. Note that $|x|$ is not differentiable at $x = 0$.

Lasso - Why sparsity in Lasso?

It is because of L_1-penalty. If L_2-penalty is used, then it is a ridge regression:

$$\text{minimize} \, \frac{1}{2N}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p}|\beta_j|^2 \leq t^2$$

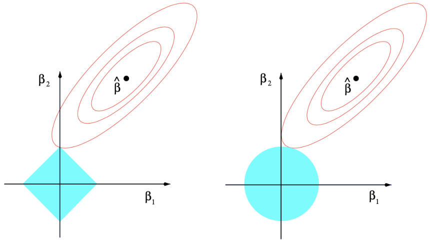

Figure 2.2 *Estimation picture for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions* $|\beta_1|+|\beta_2| \leq t$ *and* $\beta_1^2+\beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the residual-sum-of-squares function. The point* $\widehat{\beta}$ *depicts the usual (unconstrained) least-squares estimate.*

Lasso - Generalization

We can consider L_q-penalty:

$$\tilde{\boldsymbol{\beta}} = \arg\min \left\{ \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \quad \text{for } q \geq 0$$

- $q = 0$; Variable selection, count the # of non-zero coefficients
- $q = 1$; LASSO
- $q = 2$; Ridge regression
- $q > 1$; $|\beta_j|^q$ is differentiable at 0, so no sparsity

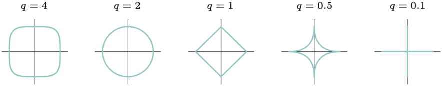

FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

- Mixture between L_1 and L_2-norm " Elastic net "

$$\lambda \sum_{j=1}^{p} (\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$$

Lasso - Coordinate descent algorithm

When λ is given, the coordinate descent algorithm updates $\beta_j(\lambda)$ one by one, iteratively.

Step1 $\tilde{\beta}_k(\lambda)$ be the current estimate for β_k. Then, rewrite penalty as

$$\frac{1}{2}\sum_{i=1}^{N}\left(y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k(\lambda) - x_{ij}\beta_j\right)^2 + \lambda\sum_{k\neq j}|\beta_k| + \lambda|\beta_j|$$

Step2 Iterate untill convergence by solving

$$0 = \sum_{i=1}^{N}\left(y_i - \sum_{k\neq j} x_{ik}\tilde{\beta}_k(\lambda) - x_{ij}\beta_j\right)(-x_{ij}) + \lambda \mathrm{sign}(\beta_j)$$

gives

$$\tilde{\beta}_j(\lambda) \leftarrow S\left(\sum_{i=1}^{N} x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda\right),$$

where $S(t, \lambda) = \mathrm{sign}(t)(|t| - \lambda)$ is a soft-thresholding operator.

Lasso - Coordinate descent algorithm

Why? Consider single-variable case.

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^{N} (y_i - x_i\beta)^2 + \lambda|\beta|$$

Need to solve "derivative = 0"

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - x_i\beta)(-x_i) + \lambda \text{sign}(\beta) = 0$$

$$\frac{1}{N} \sum_{i=1}^{N} y_i x_i - \left(\frac{1}{N} \sum x_i x_i \right) \beta - \lambda \text{sign}(\beta) = 0$$

$$\therefore \beta = \frac{1}{N} \sum_{i=1}^{N} y_i x_i - \lambda \text{sign}(\beta)$$

Lasso - Coordinate descent algorithm

Therefore, the solution becomes

$$
\begin{aligned}
\widehat{\beta} &= \begin{cases} \frac{1}{N}\sum_{i=1}^{N} y_i x_i - \lambda & \text{if } \frac{1}{N}\langle y, x\rangle > \lambda \\ 0 & \text{if } |\frac{1}{N}\langle y, x\rangle| \leq \lambda \\ \frac{1}{N}\sum_{i=1}^{N} y_i x_i + \lambda & \text{if } \frac{1}{N}\langle y, x\rangle < -\lambda \end{cases} \\
&= S\left(\frac{1}{N}\langle y, x\rangle, \lambda\right)
\end{aligned}
$$

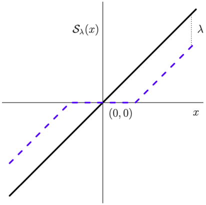

Figure 2.4 *Soft thresholding function $\mathcal{S}_\lambda(x) = \text{sign}(x)\,(|x| - \lambda)_+$ is shown in blue (broken lines), along with the 45° line in black.*

Lasso - ADMM

Alternating direction method of multipliers (ADMM) is another way of solving lasso. ADMM solves

$$\underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

subject to $\boldsymbol{\beta} - \boldsymbol{\alpha} = 0$. It has the following explicit formula:

$$\begin{cases} \boldsymbol{\beta}^{(k)} & = (X^T X + pI)^{-1}\left(X^T\mathbf{y} + p(\boldsymbol{\alpha}^{(k-1)} - \mathbf{w}^{(k-1)})\right) \\ \boldsymbol{\alpha}^{(k-1)} & = S_{\frac{\lambda}{p}}(\boldsymbol{\beta}^{(k)} + \mathbf{w}^{(k-1)}) \\ \mathbf{w}^{(k)} & = \mathbf{w}^{(k-1)} + \boldsymbol{\beta}^{(k)} - \boldsymbol{\alpha}^{(k)} \end{cases}$$

- ADMM converges very fast in a handful of iterations, but precise estimaiton requires more time than coordinate descent algorithm. See Boyd et al. (2011) for further references.

Lasso - Tuning parameter λ selection

Bootstrap and Information Criteria are most widely used.
Bootstrap (k-fold Cross validation) procedures are

1. Fit lasso with a wide range of values $\Lambda = \{\lambda_e\}_{e=1}^{m}$
2. Divide whole sample into k groups at random
3. With kth group (test set) out, fit lasso path to the remaining $k-1$ groups (training set).
4. For each $\lambda \in \Lambda$, compute mean-squared prediction error for test set.
5. Average these errors to obtain a prediction error curve
6. $\hat{\lambda}_{CV}$ is the one minimizes a prediction error curve.

Lasso - Inference on β

In order to obtain the sampling distribution of $\hat{\boldsymbol{\beta}}(\lambda)$, apply bootstrap. That is obtain subsample $\{(x_i^*, y_i^*)\}_{i=1}^N$ from the sample data $\{(x_i, y_i)\}_{i=1}^N$, and obtain lasso estimates $\{\beta^*\}$.

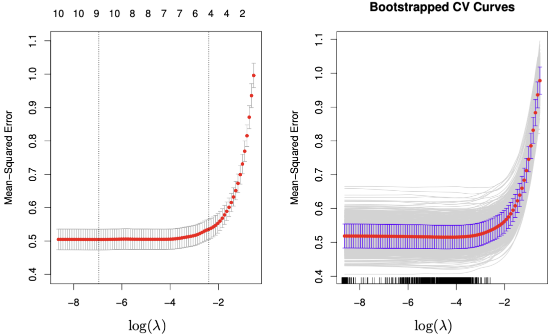

Figure 6.5 *[Left] Cross-validation curve for lasso on the diabetes data, with one-standard-error bands computed from the 10 realizations. The vertical line on the left corresponds to the minimizing value for λ. The line on the right corresponds to the* one-standard-error *rule; the biggest value of λ for which the CV error is within one standard error of the minimizing value. [Right] 1000 bootstrap CV curves, with the average in red, and one-standard-error bands in blue. The rug-plot at the base shows the locations of the minima.*

Lasso - Tuning parameter λ selection

However, CV selection may not feasible when $p \gg N$ or time dependent case, etc. Alternatively, we can use information criteria.

$$BIC = \log\left(\frac{SSE}{N}\right) + \underset{\substack{\uparrow \\ \text{\# of parameters}}}{|S_\lambda|}\frac{\log N}{N} \times C_N$$

- $C_N = 1$; usual BIC, work with moderate dimension.
- $C_N = \log(\log p)$; if $p \gg N$.

Simulations study advocate the use of BIC, but CV with $k = 10$ seems to be more popular in practice.

Lasso - Theoretical Results

Denote $\boldsymbol{\beta}^*$: true, $\widehat{\boldsymbol{\beta}}$: Lasso estimator.

1. MSE consistency

$$\frac{1}{N}\|X(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2 \leq c\|\boldsymbol{\beta}^*\|_1\sqrt{\frac{\log p}{N}}$$

   True parameter must be sparse relative to $\frac{N}{\log(p)}$.

2. Sparsistency (support recovery)

$$P\left(\mathrm{supp}(\widehat{\boldsymbol{\beta}}) = \mathrm{supp}(\boldsymbol{\beta}^*)\right) \to 1,$$

   where $\mathrm{supp}(\hat{\boldsymbol{\beta}}) = \{i : \hat{\boldsymbol{\beta}}_i \neq 0\}$ is non-zero parameters.

Lasso - Problems?

See the lasso estimator again

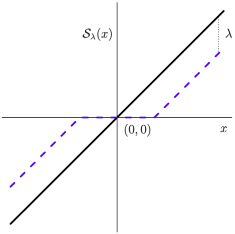

Figure 2.4 *Soft thresholding function* $\mathcal{S}_\lambda(x) = \text{sign}(x)\,(|x| - \lambda)_+$ *is shown in blue (broken lines), along with the* 45° *line in black.*

Since $\hat{\beta}$ shrinks toward zero, it introduces "bias".

Lasso - Debiasing I

Method1 Since $\hat{\boldsymbol{\beta}}^{Lasso}$ is sparsistency, estimate parameters with zero constraint. This is the best, but if p is large, it takes too much time & may not feasible sometimes.

Method2 Shrink less for large coefficients! Zou (2006) proposed the **adaptive lasso** given by

$$\hat{\boldsymbol{\beta}}^{adapt} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} w_j|\beta_j|,$$

where $\{w_j\}$ is weights for β_j. Zou (2006) suggested to use

$$w_j = \frac{1}{|\hat{\beta}_j^{init}|}, \quad j = 1, \ldots, p.$$

It is based on the weight least squares, and if $w_j = 1$, then it is a usual lasso.

Lasso - Debiasing II

Method3 Use **Non-convex penalties**. If $q < 1$, then it gives much sparse model, hence small bias.

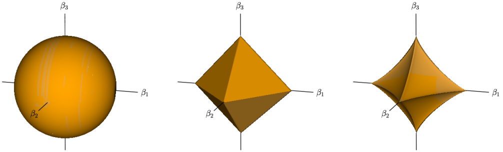

Figure 4.12 *The ℓ_q unit balls in $\mathbb{R}^3$ for $q = 2$ (left), $q = 1$ (middle), and $q = 0.8$ (right). For $q < 1$ the constraint regions are nonconvex. Smaller q will correspond to fewer nonzero coefficients, and less shrinkage. The nonconvexity leads to combinatorially hard optimization problems.*

However, it is computationally challenging if dimension p is high. Alternatively, other non-convex penalties are suggested:

- MC+ (minimax convex) by Zhang et al. (2010)
- SCAD by Fan and Li (2001)

used alternative nonconvex penalties

Lasso - Debiasing III

It can be written as

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} p(|\beta_j|),$$

where $\lambda p(\cdot)$ is a penalty function given by

$$\lambda p(t) = \int_0^t \left(I(t \le \lambda) + \frac{r\lambda - x}{(r-1)\lambda} I(t > \lambda) \right) dx \quad (SCAD)$$

$$\lambda p(t) = \int_0^t \left(1 - \frac{x}{r\lambda}\right)_+ dx \quad (MC+)$$

Lasso - Debiasing IV

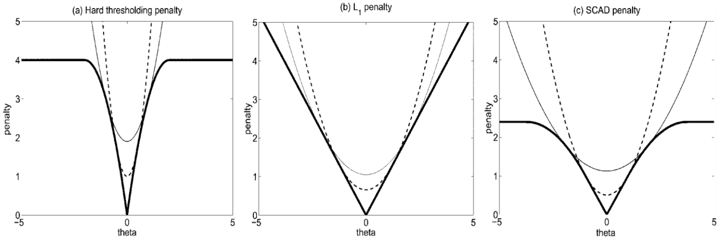

Figure 1. Three Penalty Functions $p_\lambda(\theta)$ and Their Quadratic Approximations. The values of λ are the same as those in Figure 5(c).

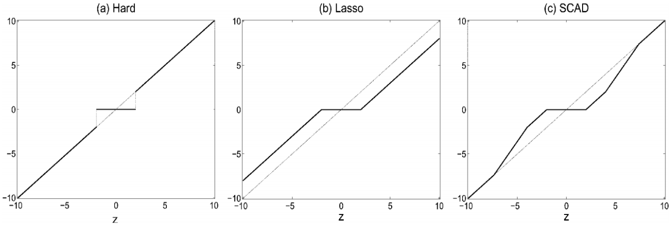

Figure 2. Plot of Thresholding Functions for (a) the Hard, (b) the Soft, and (c) the SCAD Thresholding Functions With $\lambda = 2$ and $a = 3.7$ for SCAD.

Lasso - Debiasing V

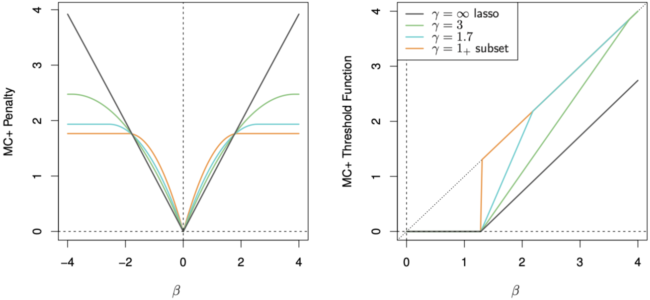

Figure 4.13 *Left: The MC+ family of nonconvex sparsity penalties, indexed by a sparsity parameter $\gamma \in (1, \infty)$. Right: piecewise-linear and continuous threshold functions associated with MC+ (only the north-east quadrant is shown), making this penalty family suitable for coordinate descent algorithms.*

Still, computationally not easy though.

Lasso - Debiasing VI

Method4 Directly subtract bias. **Debiased lasso** is defined as

$$\widehat{\boldsymbol{\beta}}^{d} = \hat{\boldsymbol{\beta}}_{\lambda} + \frac{1}{N}\Theta X'(\mathbf{y} - X\widehat{\boldsymbol{\beta}}_{\lambda}),$$

where $\widehat{\boldsymbol{\beta}}_{\lambda}$ is a standard lasso estimator and Θ is the (approximation) inverse of $\widehat{\Sigma} = 1/NX'X$. Why it works?

$$\widehat{\boldsymbol{\beta}}^{d} = \boldsymbol{\beta} + \frac{1}{N}\Theta X'\boldsymbol{\epsilon} + \underbrace{\left(I_p - \frac{1}{N}\Theta X'X\right)(\hat{\boldsymbol{\beta}}_{\lambda} - \boldsymbol{\beta})}_{=:\Delta(bias)}$$

If Θ is close enough to $N^{-1}X'X$, then $\Delta \to 0$.

$$\therefore \hat{\boldsymbol{\beta}} \sim N\left(\boldsymbol{\beta}, \frac{\sigma^2}{N}\Theta\hat{\Sigma}\Theta'\right)$$

High dimensional time series

- We will generally be interested in models for continuous-valued multivariate time series data:

$$X_t = \begin{pmatrix} X_{1,t} \\ X_{2,t} \\ \vdots \\ X_{K,t} \end{pmatrix} = (X_{j,t}), \tag{3}$$

  for $j = 1, \ldots, K, t = 1, \ldots, T$.
- We are interested in both temporal dependence (across t), and component dependence (across j).
- Data examples include functional MRI (fMRI) for brain connectivity, macroeconomic series such as gross domestic product, personal consumption expeditures, private domestic investments, financial series, stock indices etc.

High-dimensional VAR(p)

- Recall linear regression representation of VAR(p) process:

$$(x_{p+1},\ldots,x_T) = (\Phi_1 x_p + \ldots + \Phi_1 x_1, \Phi_1 x_{p+1} + \Phi_p x_2$$

$$,\ldots, \Phi_1 X_{T-1} + \cdots + \Phi_p X_{T-p}) + (z_{p+1},, ..., z_T)$$

$$= (\Phi_1\ \Phi_2\ \cdots\ \Phi_p) \begin{pmatrix} x_p & x_{p+1} & x_{p+2} & \cdots & x_{T-1} \\ x_{p-1} & x_p & x_{p+1} & \cdots & x_{T-2} \\ \vdots & \vdots & x_0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1 & x_2 & x_3 & \ldots & x_{T-p} \end{pmatrix} + (z_{p+1}, ..., z_T)$$

$$\boxed{Y = AL + Z,} \tag{4}$$

where Y is $K \times (T-p)$, A is a $K \times Kp$ parameter matrix, L is $Kp \times (T-p)$ design matrix and Z is $K \times (T-p)$ error matrix.

sparse VAR

Then, applying vec operation gives

$$\begin{aligned}\text{vec}(Y) &= \text{vec}(AL + Z)\\ &= \text{vec}(AL) + \text{vec}(Z)\\ &= \text{vec}(I_K AL) + \text{vec}(Z)\\ &= (L' \otimes I_K)\text{vec}(A) + \text{vec}(Z)\end{aligned}$$

$$\boxed{y = (L' \otimes I_K)\alpha + z,} \tag{5}$$

where $z \sim MVN(0, (I_{T-p} \times \Sigma_z))$. Denote parameter vectors

$$y = \text{vec}(Y), \quad z = \text{vec}(Z), \quad \alpha := \text{vec}(A) = \begin{pmatrix}\text{vec}(\Phi_1)\\ \text{vec}(\Phi_2)\\ \vdots\\ \text{vec}(\Phi_p)\end{pmatrix}.$$

sparse VAR

- Hence, we can apply Lasso by considering

$$\widehat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \left\{ \|y - (L' \otimes I_K)\alpha\|_2^2 + \lambda \|\alpha\|_1 \right\}.$$

- However, it does not take into account (spatial) dependence Σ_z into account.
- We wil use GLS approach here. Note that Σ_z^{-1} is symmetric so that $\Sigma_z^{-1} = \Sigma_z^{-1/2}\Sigma_z^{-1/2}$. Hence,

$$(I_{T-p} \otimes \Sigma_z^{-1/2})y = (I_{T-p} \otimes \Sigma_z^{-1/2})(L' \otimes I_K)\alpha + (I_{T-p} \otimes \Sigma_z^{-1/2})z$$

$$\iff \widetilde{y} = (L' \otimes \Sigma_z^{-1/2})\alpha + \widetilde{z},$$

where $\widetilde{z} \sim MVN(0, I_{T-p})$.

sparse VAR

- There, we can find lasso estimator from

$$\widehat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \left\{ \|\tilde{y} - (L' \otimes \Sigma_z^{-1/2})\alpha\|_2^2 + \lambda\|\alpha\|_1 \right\}$$

- Iterative algorithm to find lasso solution for sparse VAR. Note that once Σ_z is given, we can apply usual lasso algorithm.

step1 Set an inital value $\Sigma_{z,0}$, say from full VAR(p).

step2 Update coefficients α and Σ_z till convergence:

$$\widehat{\alpha}^{(k+1)} = \underset{\alpha}{\operatorname{argmin}} \left\{ \|\tilde{y} - (L' \otimes \Sigma_{z,k}^{-1/2})\alpha\|_2^2 + \lambda\|\alpha\|_1 \right\}$$

$$\Sigma_{z,k+1} = \frac{1}{T}(Y - A^{(k+1)}L))(Y - A^{(k+1)}L))'$$

- Comprehensive theoretical results are provided in Basu and Michailidis (2015).

sparse VAR: simulation

- Remark that we can apply other variants of lasso exactly to sVAR since it can be represented as lienar regression form.
- Small simulation result for VAR(1) with dimension $K = 6$ and $T = 500$.

	MLE	Lasso	DB-Lasso	Alasso	DB-Alasso
$Bias^2$	0.041	0.729	0.0484	0.093	0.047
MSE	14.22	5.99	14.06	2.367	13.96

- Sparse modeling gives better model interpretation, numerical stability, and improve forecasting.

sparse VAR: simulation

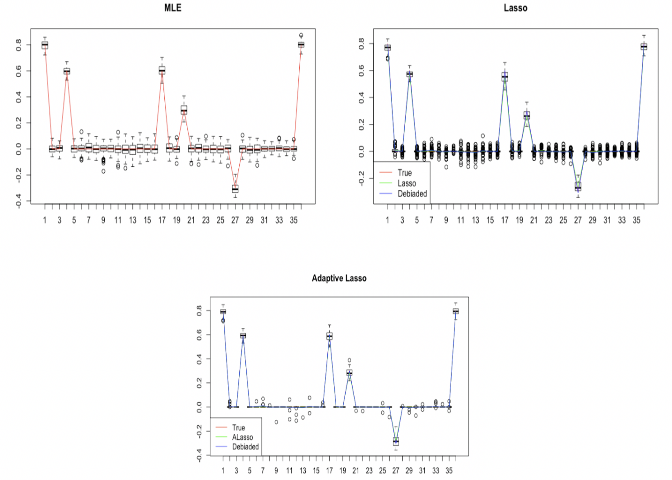

Figure: sVAR(1) simulation.

sVAR: flu data

- Google flu trend data on the weekly predicted number of influenza-like-illness (ILI) related visits per 100,000 outpatients in a US region. Google flu data is available for the 50 states, the District of Columbia and 122 major cities over the US. Used 50 states and DC.

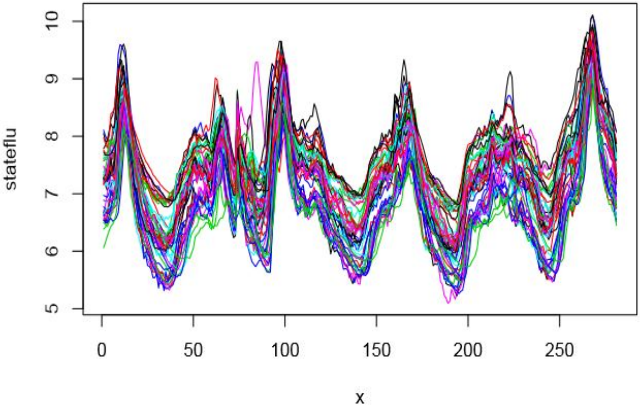

Figure: Monthly Google flu trend data for 52 states in US.

sVAR: flu data

VAR(2) model $Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \epsilon_t$ gives total $2 * 51^2 = 5202$ parameters to estimate.

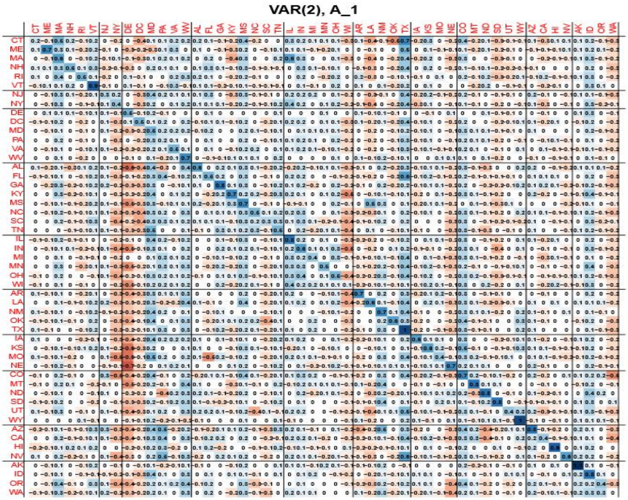

Figure: A_1

sVAR: flu data

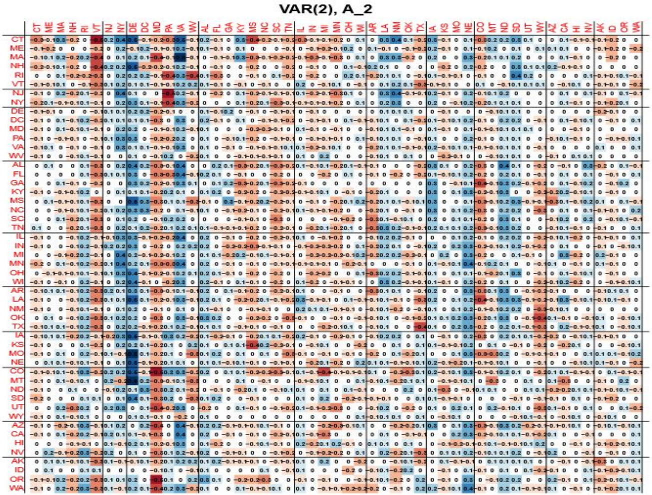

Figure: A_2

sVAR: flu data

Sparse modelling, that is set exact zero on AR coefficients on A_1, A_2 ensure better interpretation of models, numerical stability, improve prediction. For example,

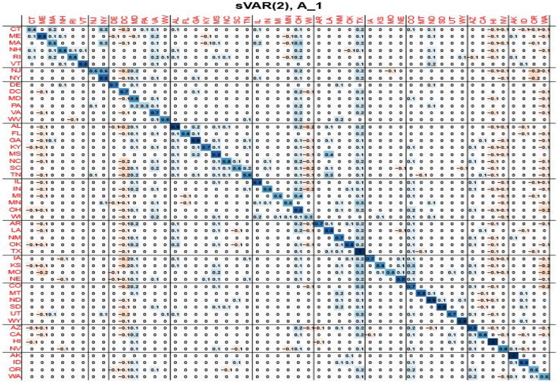

Figure: A_1

sVAR: extension to seasonal data

Cyclic variations are a.e in time series analysis! For example,

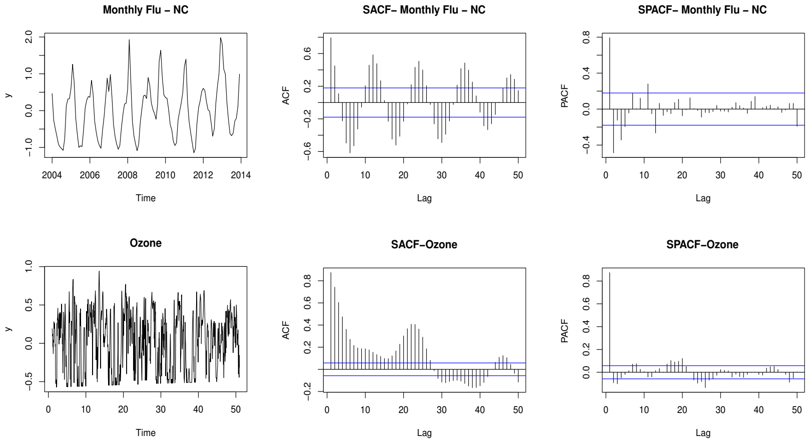

Figure: Top: Monthly flu trend in NC. Bottom: 23 hour ozone levels at a CA location. Respective sample ACFs and PACFs are given.

SVAR and PVAR

Two popular approaches in modeling seasonality:

- SVAR(P, p) (seasonal VAR)

$$\Phi(B)\Phi_s(B^s)(X_n - \mu) = \epsilon_n, \quad n \in \mathbb{Z}, \tag{6}$$

  where $\Phi(B) = 1 - A_1 B - \ldots - A_p B^p$ and $\Phi_s(B^s) = 1 - A_{s,1}B^s - \ldots - A_{s,P}B^{Ps}$ with s denotes the period.

- PVAR(p) (periodic VAR)

$$\Phi_m(B)(X_n - \mu_m) = \epsilon_{m,n}, \quad n \in \mathbb{Z}, \tag{7}$$

  where $\Phi_m(B) = 1 - A_{m,1}B - \ldots - A_{m,p}B^p$ with $A_{m,1}, \ldots, A_{m,p}$ which depend on the season $m = 1, \ldots, s$ wherein the time n falls.

sSVAR and sPVAR

- Baek et al. (2015) extended lasso algorithm to SVAR and PVAR.
- For example, adaptive LASSO for PVAR is straightforward by applying it to each season. At m-th season, corresponding coeffcient is calcualted from

$$\widehat{\boldsymbol{\beta}}_m^{(\ell)} = \underset{\boldsymbol{\beta_m}}{\operatorname{argmin}} \left(\frac{1}{T} \| (I_T \otimes \Sigma_{(\ell)}^{-1/2}) \mathbf{y}_m - (\mathbf{U}_m' \otimes \Sigma_{(\ell)}^{-1/2}) \boldsymbol{\beta}_m \|^2 + \lambda_\ell \sum_{j=1}^{p_m q^2} w_j^{(\ell)} |\beta_{m,j}| \right)$$

The covariance matrix is obtained as

$$\widehat{\Sigma}_{(\ell)} = \frac{1}{T} (\mathbf{Y}_m - \widehat{\mathbf{B}}_m^{(\ell-1)} \mathbf{U}_m)(\mathbf{Y}_m - \widehat{\mathbf{B}}_m^{(\ell-1)} \mathbf{U}_m)'.$$

Real data example: Air quality

- Air quality (CO, No, NO_2, Ozone and Solar radiation) observed hourly at Azusa, CA in 2006.
- Before fitting model, detrended with cubic polynomial regression and took log-transformation.
- The best model is selected based on h-step ahead forecast MSE. (out of sample forecasting)

$$\mathrm{MSE}(h) = \frac{1}{q(T_t - h + 1)} \sum_{t=T}^{T+T_t-h} (\widehat{Y}_{t+h} - Y_{t+h})'(\widehat{Y}_{t+h} - Y_{t+h}),$$

Real data example: Air quality

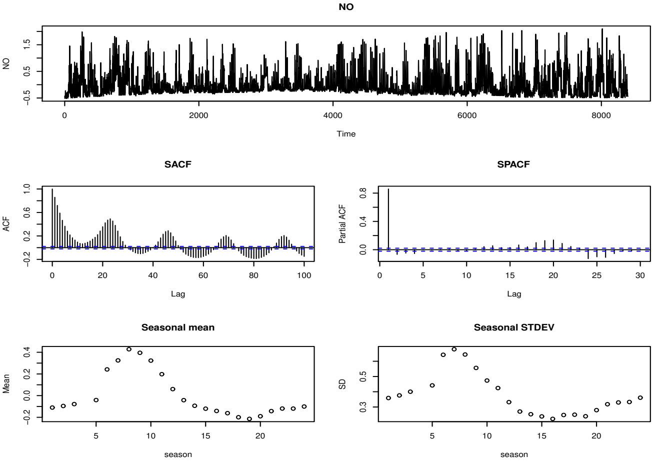

Figure: Time plot and sample ACF and PACF plots for detrended NO concentration. Seasonal mean and standard deviation are also depicted.

Real data example: Air quality

	$h=1$	$h=2$	$h=4$	$h=8$	$h=12$	$h=13$
sparse VAR(5;70)	.201	.678	2.107	6.458	8.840	9.589
PVAR(1;575)$_{23}$	.189	.273	.356	.280	.269	.270
sparse PVAR(1;320)$_{23}$ (A-LASSO)	.182	.249	.249	.238	.235	.232

Table: The h–step forecast MSE for air quality data with sparse VAR, (non-sparse) PVAR and sparse PVAR models. The sparse PVAR(1;256)$_{23}$ model achieves the smallest h–step forecast MSE in all cases considered.

Real data example: Google flu trend

- Here, we consider only 5 states (CA, GA, IL, NJ, TX) for illustration.
- Considered monthly data and take a log transformation to make the series more stationary.

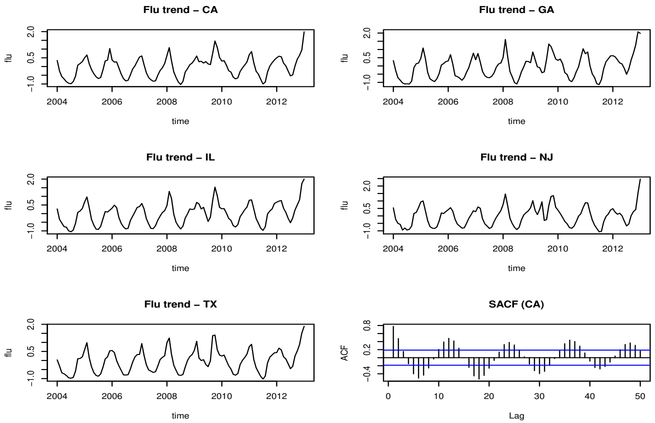

Figure: Monthly Google flu trend data.

Real data example: Google flu trend forecasting

	$h=1$	$h=2$	$h=3$
sparse VAR(1;16)	.370	.573	.813
SVAR(1,1;50)$_{12}$	.241	.396	.439
sparse SVAR(1,1;41)$_{12}$ (A-LASSO)	.222	.360	.355

Table: The h–step forecast MSE for monthly flu data with sparse VAR, (non-sparse) SVAR and sparse SVAR models.

References

- Basu, S., & Michailidis, G. (2015). Regularized estimation in sparse high-dimensional time series models. The Annals of Statistics, 43(4), 1535-1567.
- Baek, C. Davis, R. A., Pipiras, V. (2017). Sparse seasonal and periodic vector autoregressive modeling. Computational Statistics and Data Analysis 106 (2017) 103–126.
- Davis, R. A., Zang, P., and Zheng, T. (2015). Sparse vector autoregressive modeling. Journal of Computational and Graphical Statistics.
- Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity: the lasso and generalizations. CRC press.
- Zou, H. (2006). Adaptive lasso and Its Oracle Properties. *Journal of American Statistical Association*, **101**, 1418–1429.