Support Recovery from Covariance Information using Non-Negative Quadratic Programming

Lekshmi Ramesh

Signal Processing for Communications Lab
IISc, Bangalore

October 30, 2016

Problem setup

- Model: Observations $\{y_i\}_{i=1}^L$ are generated from the following linear model:

$$y_i = \Phi x_i + w_i, \quad i \in [L],$$

  where $\Phi \in \mathbb{R}^{m \times N}$ $(m < N)$, x_i unknown, random taking values in $\mathbb{R}^N$ and $w_i \sim \mathcal{N}(0, \sigma^2 I)$.

- Assumptions:
  - $\text{supp}(x_i) = T$ for some $T \subset [N]$ with $|T| = k$, $\forall i \in [L]$
  - $\mathbb{E}[x_{t,i} x_{t,j}] = 0$, $t \in [L]$, $i, j \in T$

Problem Setup

- We impose the following prior on x_i

$$p(x_i;\gamma) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi\gamma_j}} \exp\left(-\frac{x_{ij}^2}{2\gamma_j}\right)$$

$$\text{i.e.,} \quad x_i \overset{iid}{\sim} \mathcal{N}(0,\Gamma) \text{ where } \Gamma = \operatorname{diag}(\gamma)$$

- Goal: Recover the common support T from $\{y_i\}_{i=1}^{L}$

- Note:
  - $\text{supp}(x_i) = \text{supp}(\gamma) = T$ (since $\gamma_j = 0 \Leftrightarrow x_{ij} = 0$ a.s.)
  - $y_i \sim \mathcal{N}(0, \underbrace{\Phi\Gamma\Phi^\top + \sigma^2 I}_{\Sigma \in \mathbb{R}^{m\times m}})$
- Equivalent problem: Recover Γ from an estimate of Σ
  - We work with the sample covariance matrix $\hat{\Sigma}$
- Express $\hat{\Sigma}$ as

$$\hat{\Sigma} = \Sigma + N,$$

where N: Noise/Error matrix.

- Equivalently (for $\sigma^2 = 0$),

$$\hat{\Sigma} = \Phi\Gamma\Phi^\top + N$$

$$\Big\downarrow \text{vectorize}$$

$$r = \underbrace{(\Phi \odot \Phi)}_{A \in \mathbb{R}^{m^2 \times N}} \gamma + n$$

($\odot$ denotes the Khatri-Rao product)

- We will find the Maximum-Likelihood estimate of γ.
  For that, we first derive the noise statistics.

Noise Statistics

- Mean

$$\mathbb{E}N = \frac{1}{L}\sum_{i=1}^{L}\mathbb{E}y_i y_i^\top - \Sigma = 0,$$

- Covariance

$$\begin{aligned}
\mathrm{cov}(N) &= \mathrm{cov}\left(\frac{1}{L}\sum_{i=1}^{L} y_i y_i^\top - \Sigma\right) \\
&= \mathrm{cov}\left(\sum_{i=1}^{L}\left(\frac{y_i y_i^\top}{L} - \frac{\Sigma}{L}\right)\right) \\
&= L\mathrm{cov}\left(\frac{y_1 y_1^\top}{L} - \frac{\Sigma}{L}\right) && \text{(sum of } L \text{ i.i.d. random matrices)} \\
&= \frac{1}{L}\mathrm{cov}(y_1 y_1^\top - \Sigma) \\
&= \frac{1}{L}\mathrm{cov}(y y^\top).
\end{aligned}$$

- Represent y as

$$y = Cz,$$

where $z \sim \mathcal{N}(0, I)$ and $\Sigma = CC^\top$.

For $\sigma^2 = 0$, we can take $C = \Phi\Gamma^{\frac{1}{2}}$

$$\begin{aligned}
\operatorname{cov}(\operatorname{vec}(N)) &= \frac{1}{L}\operatorname{cov}(\operatorname{vec}(Czz^\top C^\top)) \\
&= \frac{1}{L}\operatorname{cov}((C \otimes C)\operatorname{vec}(zz^\top)) \\
&= \frac{1}{L}(C \otimes C)\operatorname{cov}(\operatorname{vec}(zz^\top))(C \otimes C)^\top \\
&= \frac{1}{L}(\Phi \otimes \Phi)(\Gamma^{\frac{1}{2}} \otimes \Gamma^{\frac{1}{2}})\underbrace{\operatorname{cov}(\operatorname{vec}(zz^\top))}_{B \in \mathbb{R}^{N^2 \times N^2}}(\Gamma^{\frac{1}{2}} \otimes \Gamma^{\frac{1}{2}})(\Phi \otimes \Phi)^\top
\end{aligned}$$

- Last step: use $(A \otimes B)(C \otimes D) = AB \otimes CD$

Example: N=3

- Let $z = [z_1, z_2, z_3]^\top$ with $z_i \sim \mathcal{N}(0,1)$. Then,

$$zz^\top = \begin{bmatrix} z_1^2 & z_1 z_2 & z_1 z_3 \\ z_1 z_2 & z_2^2 & z_2 z_3 \\ z_1 z_3 & z_2 z_3 & z_3^2 \end{bmatrix} \xrightarrow{vectorize} \begin{bmatrix} z_1^2 \\ z_1 z_2 \\ z_1 z_3 \\ z_1 z_2 \\ z_2^2 \\ z_2 z_3 \\ z_1 z_3 \\ z_2 z_3 \\ z_3^2 \end{bmatrix}$$

- The covariance matrix B of $\mathrm{vec}(zz^\top)$ will be of size 9×9 with $B_{i,j} \in \{0, 1, 2\},\ 1 \leq i, j \leq 3$.

- For e.g.,

$$
\begin{aligned}
B_{1,1} &= \mathrm{cov}(z_1^2, z_1^2) = \mathbb{E}z_1^4 - (\mathbb{E}z_1^2)^2 = 3 - 1 = 2 \\
B_{1,2} &= \mathrm{cov}(z_1^2, z_1 z_2) = \mathbb{E}z_1^3 z_2 - \mathbb{E}z_1^2 \mathbb{E}z_1 z_2 = 0 \\
B_{2,4} &= \mathrm{cov}(z_1 z_2, z_1 z_2) = \mathbb{E}z_1^2 z_2^2 - \mathbb{E}z_1 z_2 \mathbb{E}z_1 z_2 = 1
\end{aligned}
$$

$$B = \mathrm{cov}(\mathrm{vec}(zz^\top)) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

- We now have the following model

$$r = A\gamma + n, \tag{1}$$

where

$$\begin{aligned} A &= (\Phi \odot \Phi), \\ \mathbb{E}[n] &= 0, \\ \text{cov}(n) = W &= \frac{1}{L}(\Phi \otimes \Phi)(\Gamma^{\frac{1}{2}} \otimes \Gamma^{\frac{1}{2}})B(\Gamma^{\frac{1}{2}} \otimes \Gamma^{\frac{1}{2}})(\Phi \otimes \Phi)^\top. \end{aligned}$$

Observations

- The noise term vanishes as $L \to \infty$
- The noise covariance depends on the parameter to be estimated
- r, $\Phi \odot \Phi$ and n have redundant entries – restrict to the $\frac{m(m+1)}{2}$ distinct entries

New model, Gaussian approximation

- Pre-multiply (1) by $P \in \mathbb{R}^{\frac{m(m+1)}{2} \times m^2}$, formed using a subset of the rows of I_{m^2}, that picks the relevant entries. Thus,

$$r_P = A_P \gamma + n_P,$$

where $r_P := Pr$, $A_P := PA$, and $n_P := Pn$.

- Further, we approximate the distribution of n_P by $\mathcal{N}(0, W_P)$, where $W_P = PWP^\top$

- Thus, $r_P \sim \mathcal{N}(A_P \gamma, W_P)$.

ML estimation of γ

- Denote the ML estimate of γ by γ_{ML}

$$\gamma_{ML} = \underset{\gamma \geq 0}{\arg\max}\ p(r_P; \gamma), \tag{2}$$

where

$$p(r_P; \gamma) = \frac{1}{(2\pi)^{\frac{m(m+1)}{4}} |W_P|^{\frac{1}{2}}} \exp\left(\frac{-(r_P - A_P\gamma)^\top W_P^{-1} (r_P - A_p\gamma)}{2} \right).$$

- Simplifying (2), we get

$$\gamma_{ML} = \arg\min_{\gamma \geq 0} \; \log|W_P| + (r_P - A_P\gamma)^\top W_P^{-1}(r_P - A_p\gamma). \qquad (3)$$

- To solve (3) (recall W_P depends on γ):
  - Initialize W_P
  - Solve
  $$\arg\min_{\gamma \geq 0} \; (r_P - A_P\gamma)^\top W_P^{-1}(r_P - A_p\gamma)$$
  - Recompute W_P and iterate

Non-negative quadratic program

$$\underset{\gamma \geq 0}{\text{minimize}} \;\; (r_P - A_P \gamma) W_P^{-1} (r_P - A_p \gamma)^\top$$

Solution (entry-wise update equation for γ):

$$\gamma_j^{(i+1)} = \gamma_j^{(i)} \left(\frac{-b_j + \sqrt{b_j^2 + 4(Q^+ \gamma^{(i)})_j (Q^- \gamma^{(i)})_j}}{2(Q^+ \gamma^{(i)})_j} \right),$$

where $b = -A_P^\top W_P^{-1} r_P$, $Q = A_P^\top W_P^{-1} A_P$,

$$Q_{ij}^+ = \begin{cases} Q_{ij}, & \text{if} \quad Q_{ij} > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad Q_{ij}^- = \begin{cases} -Q_{ij}, & \text{if} \quad Q_{ij} < 0, \\ 0, & \text{otherwise}. \end{cases}$$

Support recovery performance

$N = 52, m = 7, L = 30$; exact recovery over 200 trials

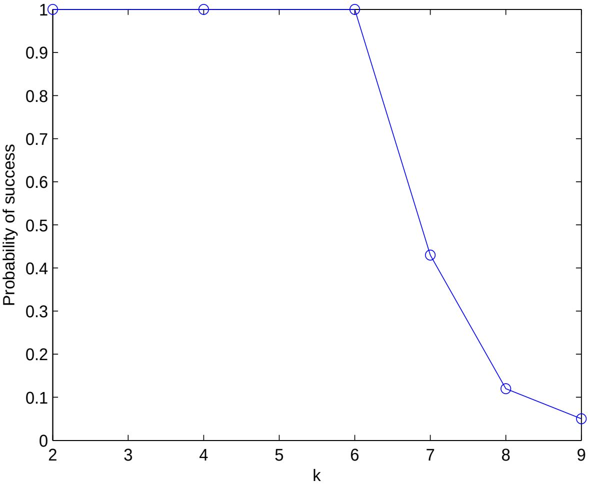

Figure 1: Support recovery performance of the NNQP-based approach

Support recovery performance

$N = 52, m = 8, L = 30$; exact recovery over 200 trials

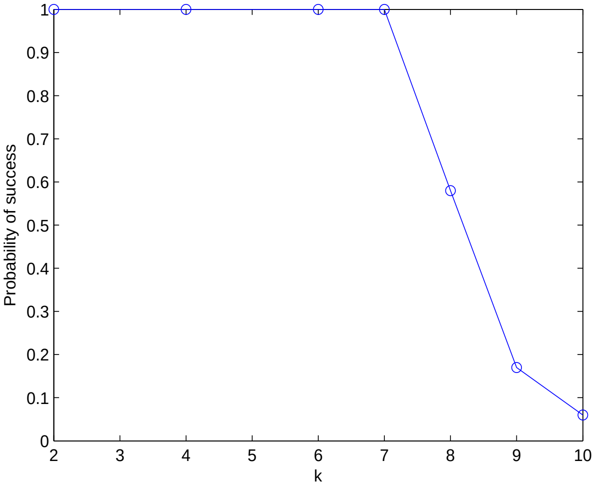

Figure 2: Support recovery performance of the NNQP-based approach

Support recovery performance

$N = 37, m = 10, k = 9$; exact recovery over 200 trials

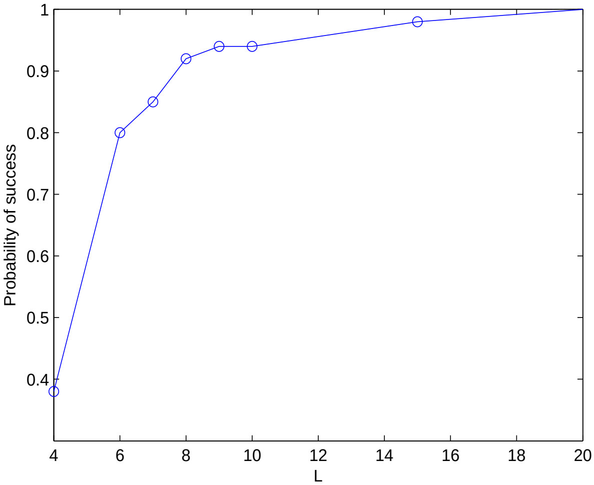

Figure 3: Support recovery performance of the NNQP-based approach

Support recovery performance

$N = 37, m = 10, k = 15$; exact recovery over 200 trials

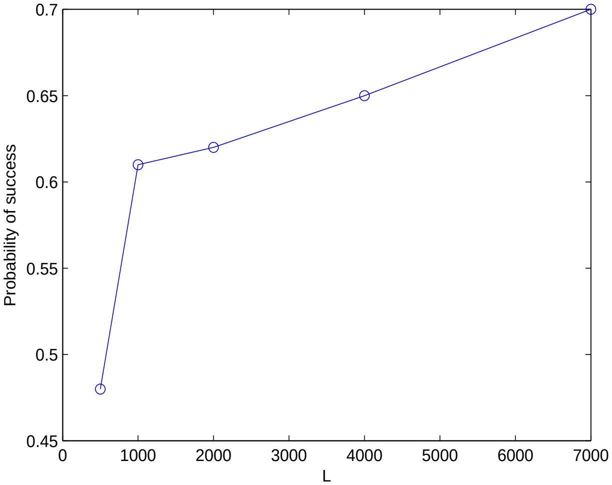

Figure 4: Support recovery performance of the NNQP-based approach

Observations, Future Work

- Observations
  - Exact support recovery possible for $k < m$ with 'small' L
  - For $m \leq k \leq \alpha m$ for some $1 \leq \alpha < \frac{N}{m}$, recovery possible with 'large' L
  - Runtime of the NNQP-based algorithm does not scale with L, but scales with m, N
- Future Work
  - Study the behavior of the Restricted Isometry Constant of the matrix A_P
  - Develop a faster algorithm for solving the maximum-likelihood problem

References

Sha, F., L. K. Saul, and D. D. Lee. "Multiplicative Updates for Nonnegative Quadratic Programming in Support Vector Machines". In: *Neural Information Processing Systems* (2002).