Graphic Lasso: Why bad estimation?

Jiaxin Hu

December 22, 2020

- Why the simulation in Matlab, Line 133 gives a bad estimation?

1 Possible Reasons

1. **Threshold for "nonzero"**

   In the original Matlab code, we use "$\sim=$" to check whether the element is equal to 0 or not, which may be too strict for "nonzero". Therefore, I use $1e^{-10}$ as the threshold for "nonzero", i.e., the element with absolute value smaller than $1e^{-10}$ is considered as 0.

2. **Tuning parameter**

   In the original Matlab code, we use function `GLasso`, which gives the results after BIC selection. However, the BIC criterion may not be good enough in this case. Figures 1, 2, and 3 plot the true Ω, and $\hat{\Omega}$ with BIC criterion and given ρ.

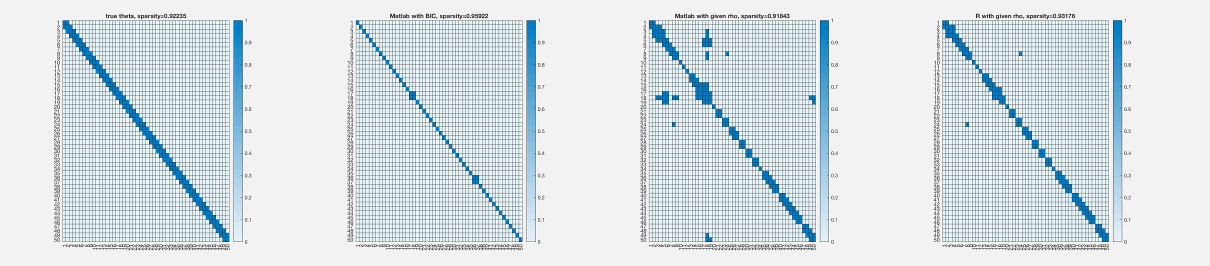


Figure 1: True Ω versus $\hat{\Omega}$ given by (1) Matlab with BIC (2) Matlab with $\rho = 0.5$, and (3) R package with $\rho = 0.5$. Here $k = 1$ corresponding to the chain network setting.

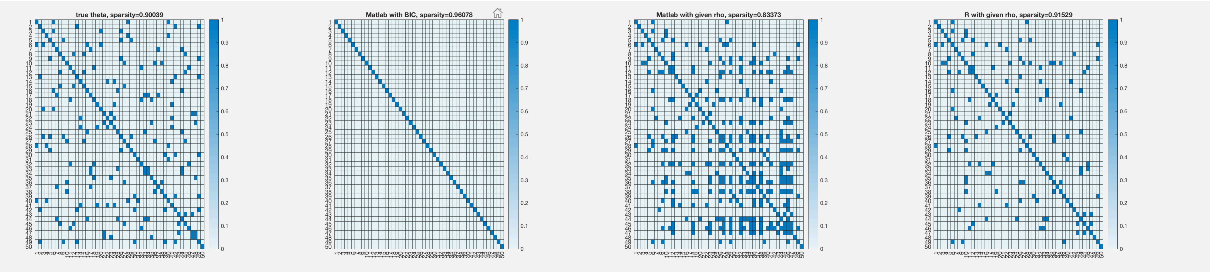


Figure 2: True Ω versus $\hat{\Omega}$ given by (1) Matlab with BIC (2) Matlab with $\rho = 0.3$, and (3) R package with $\rho = 0.3$. Here $k = 3$ corresponding to the m-nearest network setting.

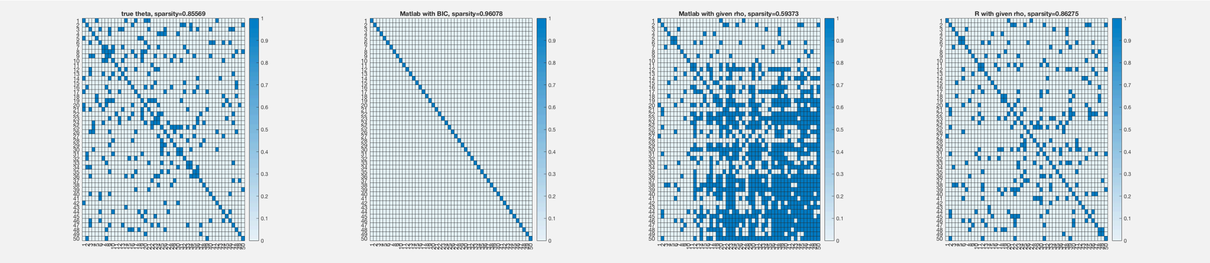


Figure 3: True Ω versus $\hat{\Omega}$ given by (1) Matlab with BIC (2) Matlab with $\rho = 0.2$, and (3) R package with $\rho = 0.2$. Here $k = 5$ corresponding to the random network setting.

From these figures, we know that BIC criterion may not be a good way to determine the ρ in this model. With given specific ρ, the Matlab and R implementations will give better results.

3. **Matlab vs R implementation**

According to above simulations, we would like to compare the Matlab and R implementation with given tuning parameter ρ and check which implementation is more sensitive to the parameter. Here we take random network as an example, i.e., $k = 5$. Figures 4, 5, 6, and 7 plot the true Ω and $\hat{\Omega}$ given by Matlab and R with different ρ.

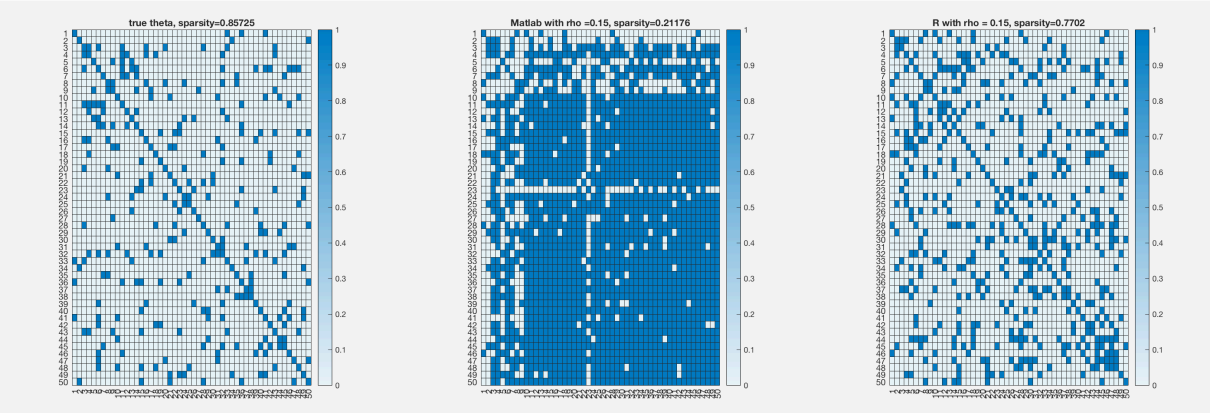
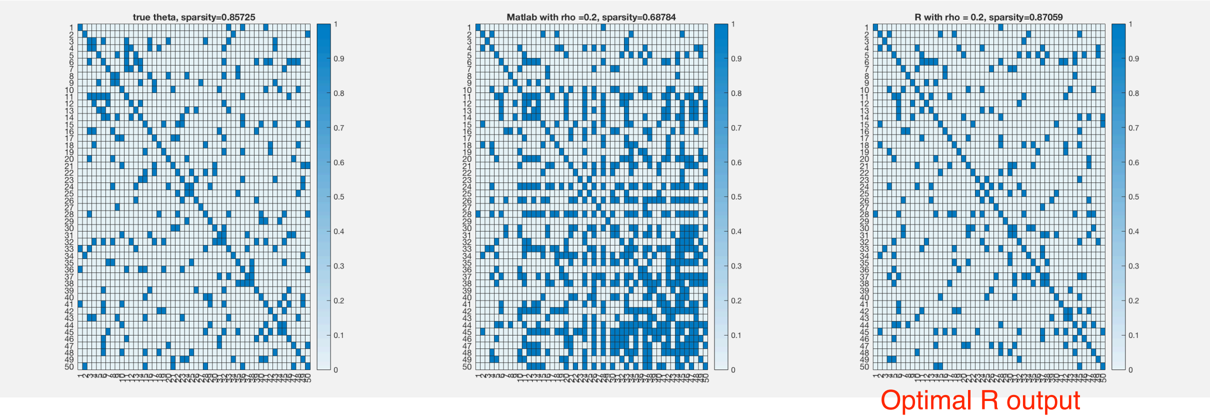


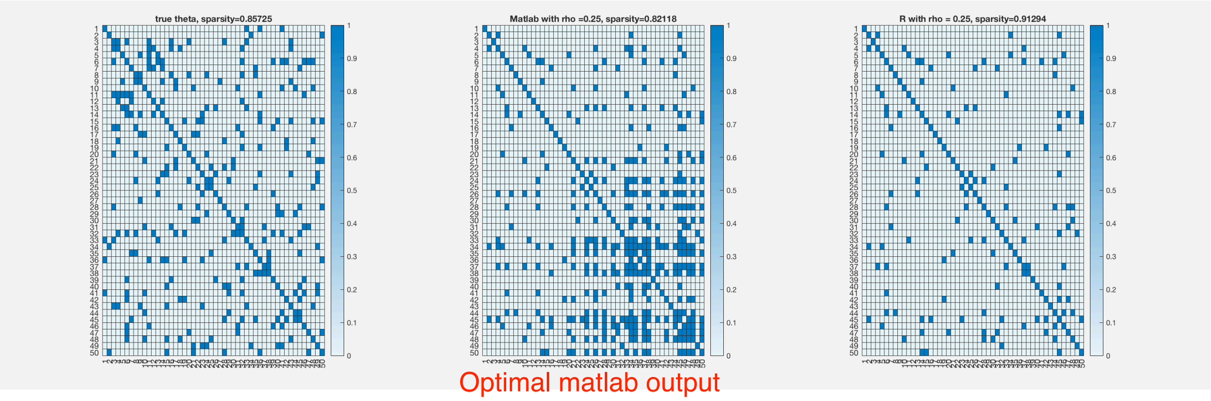
Figure 4: True Ω versus $\hat{\Omega}$ given by Matlab and R package with $\rho = 0.15$.



Optimal R output

Figure 5: True Ω versus $\hat{\Omega}$ given by Matlab and R package with $\rho = 0.2$.

Does rho have the same definitions under matlab vs. R?
Alternative visualization:
plot the optimal R output (rho~0.2), optimal matlab output (rho~0.25), vs. ground truth,
such that these three matrices all have similar sparsity ~ 0.85.
Based your figure 5 and 6, R seems to recover the pattern better. Please confirm numerically.



Optimal matlab output

Figure 6: True Ω versus $\hat{\Omega}$ given by Matlab and R package with $\rho = 0.25$.

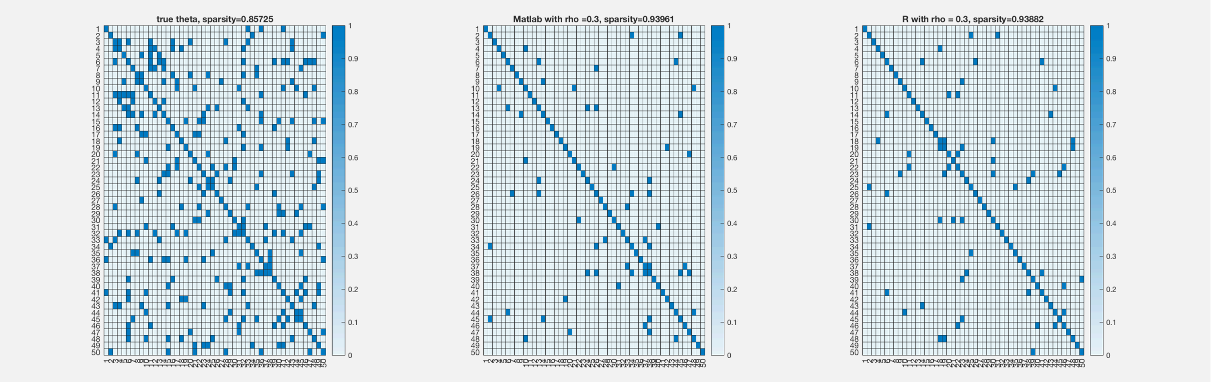


Figure 7: True Ω versus $\hat{\Omega}$ given by Matlab and R package with $\rho = 0.3$.

According to the figures, we can see that Matlab code may be more sensitive to the tuning parameter and R implementation gives better estimation with proper ρ.

Conclusion: Based on the mentioned analysis, I believe the method itself works. The threshold for "nonzero", criterion for tuning parameter selection, and the Matlab implementation affect the performance of estimation. I believe we should use the R package `glasso`.

good to know.

2 Next

Next, I would love to try ROC/AUC criterion for selecting tuning parameter.

I would defer this analysis for latter.

Next step, let's investigate the statistical properties of the model. In particular, parameter identifiability, accuracy, etc