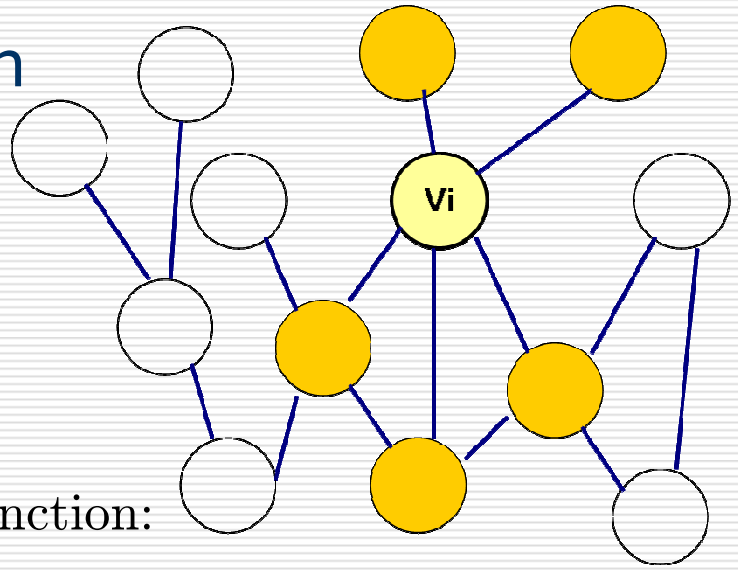
SINCO - a greedy coordinate ascent approach for sparse inverse covariance selection

Katya Scheinberg
Columbia University
(joint work with I. Rish)

The Matrix is everywhere, it is all around us, even now in this very room. You can see it when you look out your window, or you turn on your television. You can feel it when you go to work, when you go to church, when you pay your taxes. It is the world that has been pulled over your eyes to blind you from the truth.

Morpheus, "The Matrix", movie

Sparse inverse covariance selection



p random varibles

$\boldsymbol{x} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_p\}$

Multivariate Gaussian probability density function:

$$P(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- X - N realizations of $\boldsymbol{x}$, $(\mu = 0)$
- $\max_\Sigma \log(P(X)) = \max_\Sigma \frac{N}{2} \log(\det(\Sigma^{-1})) - \frac{1}{2} Tr((XX^\top)\Sigma^{-1})$
- Zeros in Σ^{-1} : conditional independence
- Sparsity $\Rightarrow$ better interpretation & prediction.

Optimizing log likelihood

- $\max_\Sigma \log(P(X)) = \max_\Sigma \frac{N}{2} \log(\det(\Sigma^{-1})) - \frac{1}{2} Tr((XX^\top)\Sigma^{-1})$
- Let $A = \frac{1}{N} XX^\top$
- $\Sigma^{-1} = \arg\max_C \frac{N}{2} (\log\det C - Tr(AC))$

  - Solution $\Sigma^{-1} = A^{-1}$ - typically not sparse.
  - Need to enforce sparsity of Σ^{-1}: Penalize for nonzeros

Enforcing sparsity

- NP-hard formulation

$$\Sigma^{-1} = \arg\max_C \left(\log\det C - Tr(AC) - \lambda Card(C)\right)$$

- Convex relaxation

$$\Sigma^{-1} = \arg\max_C \log\det C - Tr(AC) - \lambda||C||_1$$

$$\left(\|C\|_1 = \sum_{ij} |C_{ij}|\right)$$

- Convex optimization problem with unique solution for each λ

Choosing the regularization parameter λ

- λ decides the amount of sparsity.

- What is the criteria to pick the best λ?
  - Model structure recovery
  - Prediction power on test data
- Often need to compute the path of solutions for a range of λ.
- Look at the "ROC curve" – plotting the number of true positives versus false positives for different values of λ.
- Decide which value of lambda and which solution to pick.

Primal-dual pair of problems

Primal problem

$$\max_{C \succ 0} \frac{N}{2} \ln(\det(C) - Tr(AC)) - \lambda \|C\|_1$$

Dual problem

$$\max_{W \succ 0} \{ \frac{N}{2} \ln(\det(W)) - Np/2 : \text{ s.t. } -\lambda \leq \frac{N}{2}(W - A) \leq \lambda \}$$

First-order methods

Grad. Proj. by Duchi et al. – gradient projection for nonneg orthant

COVSEL, by Benarjee et al. – applying Nesterov first order method $O(1/\epsilon)$

VSM by Lu – a variant of smooth first order method $O(1/\sqrt{\epsilon})$

ADAL by Ma, Goldfarb and S. – alternating directions augmented Lagrangian $O(1/\sqrt{\epsilon})$

On each iteration compute the gradient

$$W = C^{-1}$$

$$O(p^3) \text{ operations}$$

Block coordinate ascent

Update one row and one column of the dual matrix W at each step

COVSEL, by Benarjee et al. - block coordinate descent + IPM

GLASSO by Tibshirani et al. - block coordinate descent + Coord. Des.

$$W = \begin{bmatrix} W_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$\max_{W \succ 0} \{ \tfrac{N}{2} \ln(\det(W)) - Np/2 : \text{s.t. } -\lambda \leq \tfrac{N}{2}(W - A) \leq \lambda \}$$

$$\min_{w_{12}} \{ w_{12}^{\top} W_{11}^{-1} w_{12} : \quad \text{s.t.} \quad \frac{N}{2} \| w_{12} - a_{12} \|_{\infty} \leq \lambda,$$

Subproblems reduce to LASSO regression

Work per iterations – solving Lasso regression in n variables

Coordinate descent for Lasso

$$\min_{x_i} \quad \frac{1}{2}\|Ax - b\|^2 + \lambda\|x_i\|_1$$

Choose one variable x_i and column A_i.
Let $\bar{x}$ and $\bar{A}$ correspond to the fixed part

$$\min_{x_i} \quad \frac{1}{2}(A_i x_i + \bar{A}\bar{x} - b)^2 + \lambda|x_i|$$

Soft-thresholding operator

$$\min_{x_i} \frac{1}{2}(x_i - r)^2 + \lambda|x| \rightarrow x_i = \begin{cases} r - \lambda & \text{if } r > \lambda \\ 0 & \text{if } -\lambda \leq r \leq \lambda \\ r + \lambda & \text{if } r < -\lambda \end{cases}$$

$$r = -A_i{}^\top(\bar{A}\bar{x} - b)/||A_i||, \ \lambda \rightarrow \lambda/||A_i||^2$$

Coordinate descent variants

Can choose coordinate to update by:

- Simply cycle through all coordinates (slow?)
- Update all at once (may violate pos. def. constraint)
- Choose the one with largest gradient component (slow)
- Choose the one with largest obj. function improvement!

The idea is to generate nonzero entries according the their “importance”

Coordinate descent for the original primal

$$\max_{C \succ 0} \frac{N}{2} \ln(\det(C) - Tr(AC)) - \lambda \|C\|_1$$

$\bar{C} = C + \alpha(e_i {e_j}^T + e_j {e_i}^T)$ - sparse rank-two update

KEY: simple closed form solution for optimizing and updating

$$\det(C + \alpha(e_i {e_j}^T + e_j {e_i}^T)) = \det(C)(1 + 2\alpha W_{ij} - \alpha^2(W_{ii}W_{jj} - W_{ij}^2)))$$

$$\min_{\alpha} \log(1 + 2\alpha W_{ij} - \alpha^2(W_{ii}W_{jj} - W_{ij}^2)) - 2\alpha A_{ij} - 2\lambda|C_{ij} + \alpha|$$

Can be solved by a quadratic equation

Work per iteration

For each (i,j) pair exact line search for coordinate descent:
solve a quadratic equation (solution always exists)

Update the gradient W (via a rank-two update to C):

$$\bar{W} = W - \alpha(\kappa_1 W_i {W_j}^T + \kappa_2 W_i {W_i}^T + \kappa_3 W_j {W_j}^T + \kappa_1 W_j {W_i}^T)$$

It takes O(p^2) to update W, hence we can afford to take O(p^2) to choose a good (i,j):

Consider all pairs (i,j) and choose the best obj. function improvement.

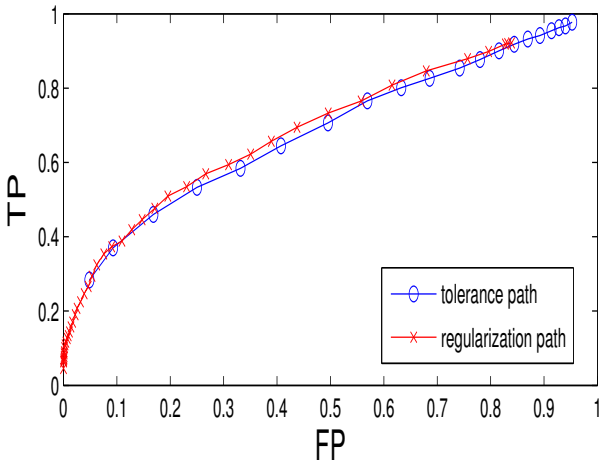
Summary of the approach

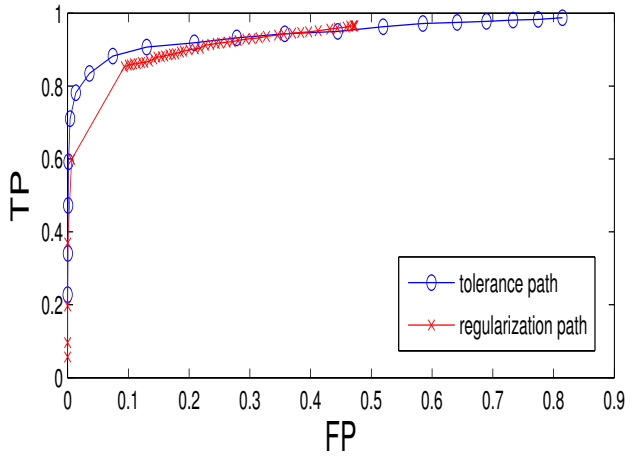
- For each pair (i,j) the exact line search takes a constant number of operations.
- In $O(p^2)$ operations we can find the step with the **best** improvement.
- In $O(p^2)$ operations W matrix is updated. Hence each iteration takes $O(p^2)$ ops.
- Can be easily parallelized.
- Sparsity is naturally preserved.

Testing environment

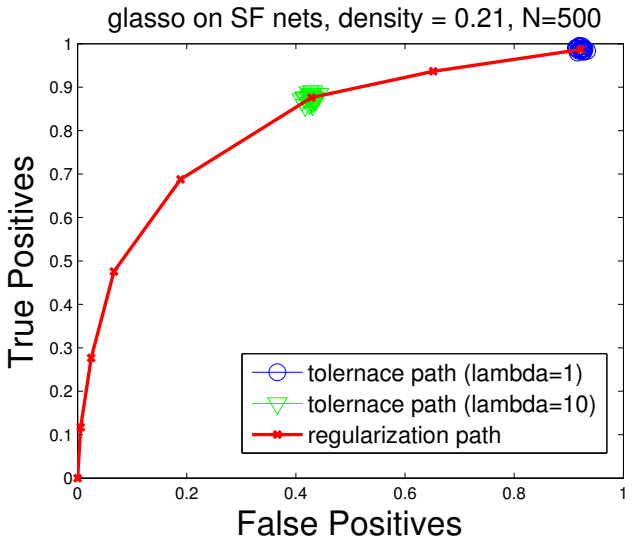
- We generated networks ($p \times p$ matrices) with various % of non-zero entries.
- These matrices define the “ground-truth'' inverse covariance matrices.
- We then computed corresponding covariance matrices.
- We then sampled n instances, from the corresponding multivariate Gaussian distribution over p variables.
- We compared out Matlab/C++ code against Glasso - Fortran/R code and COVSEL Matlab/C++ code.

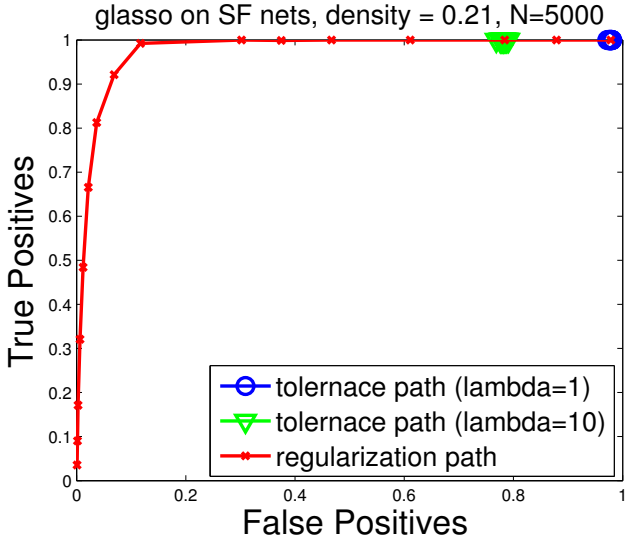
ROC curves for varying tolerance or λ in SINCO



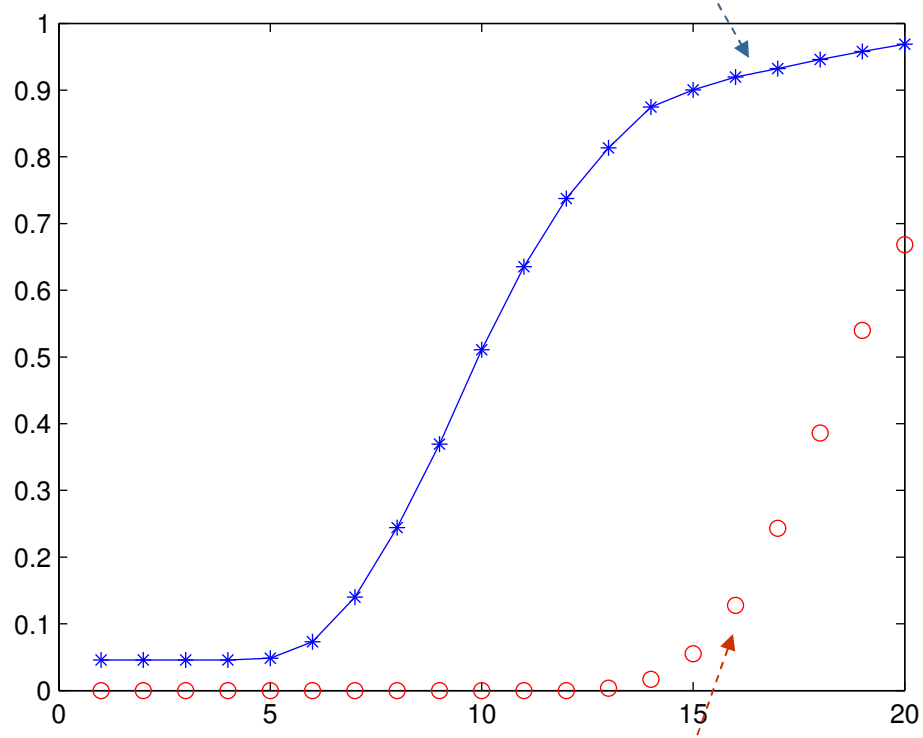


ROC curves for GLASSO



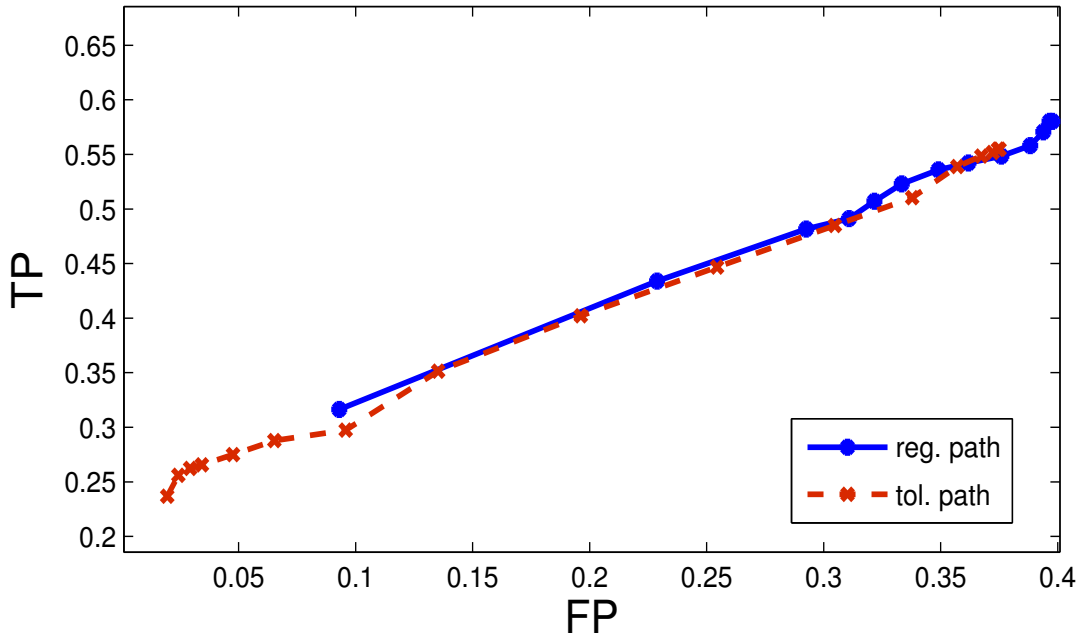


Percentage of true positives in λ paths



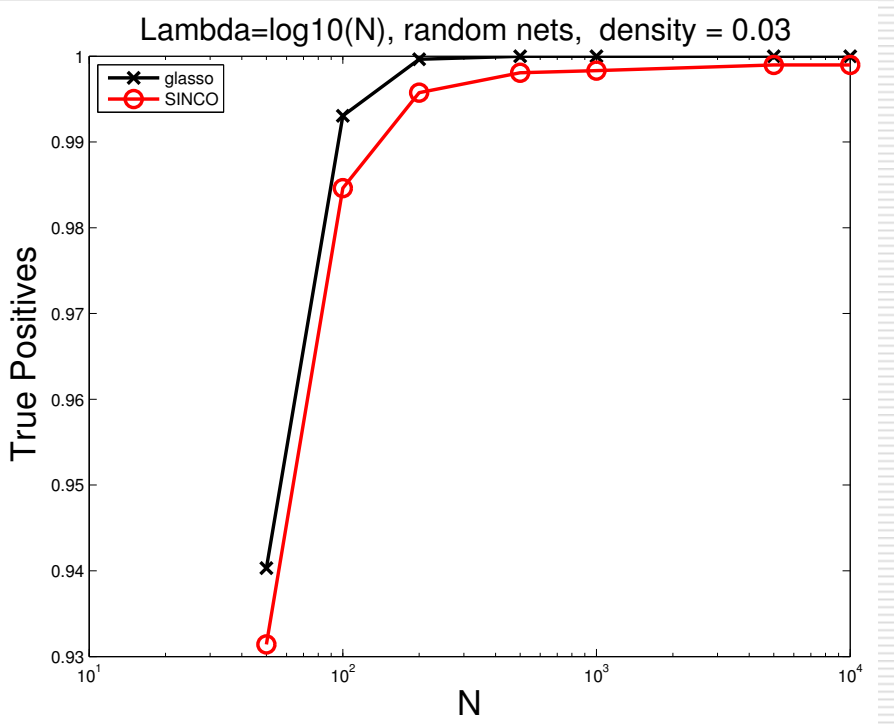
Percentage of positives in tolerance path that are not in λ path

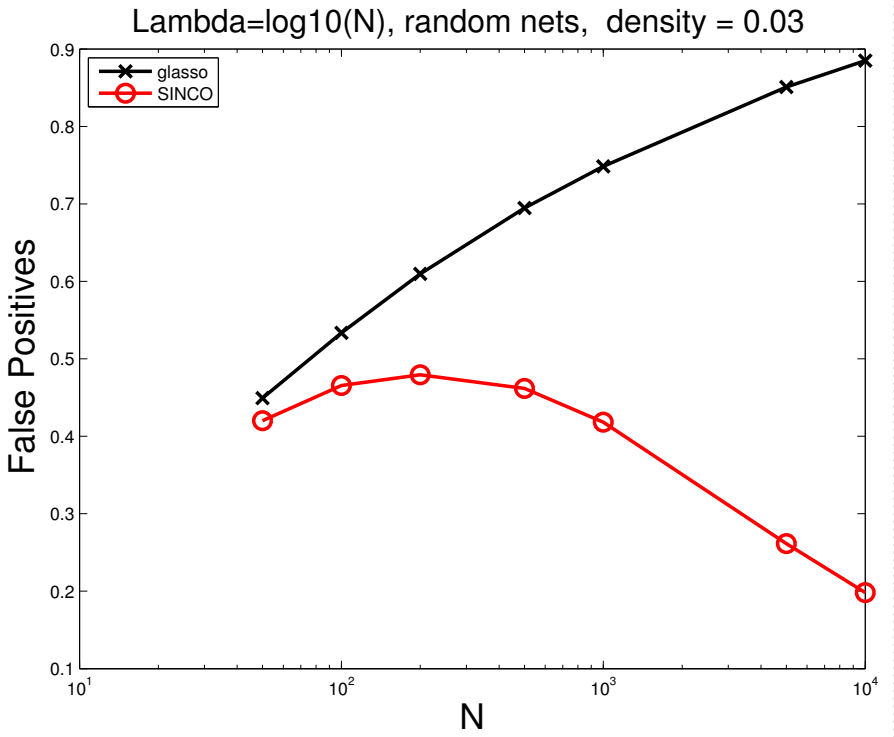
Poor ROC curves for tolerance and lambda still follow each other



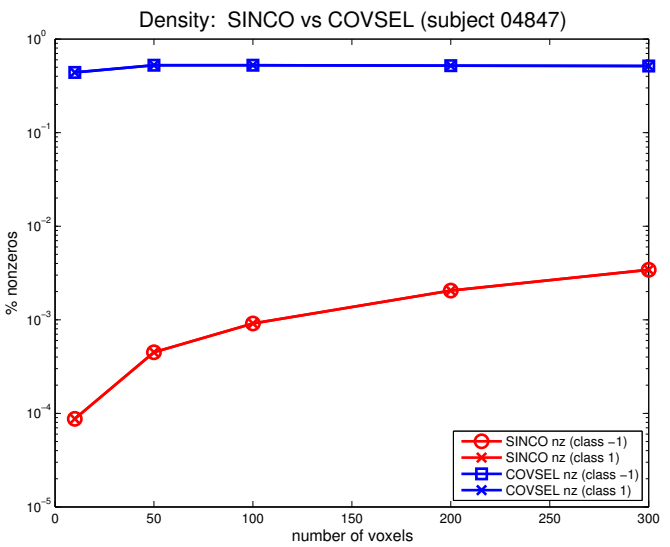
E. Coli data, p=133, n=300

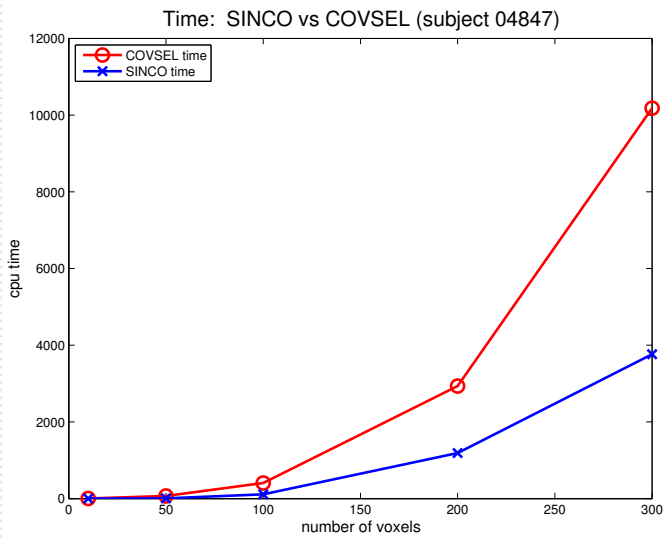
TP and FP with growing N and lambda

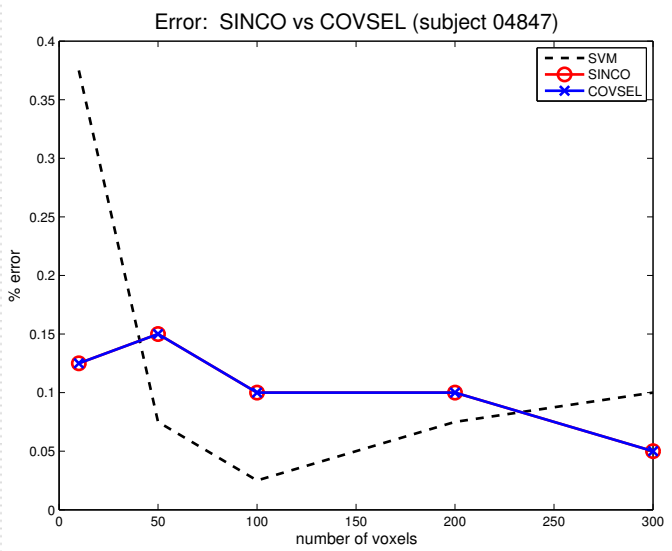




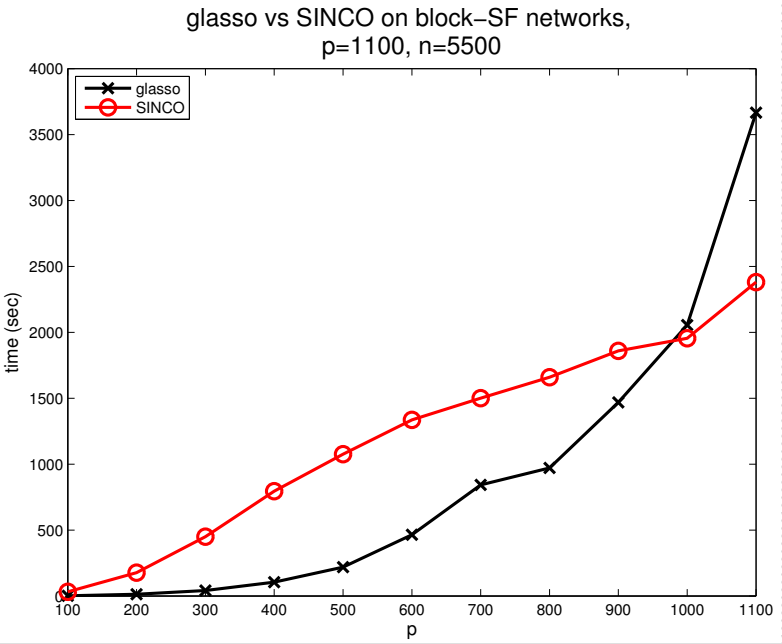
Classification task using the estimated inverse covariance

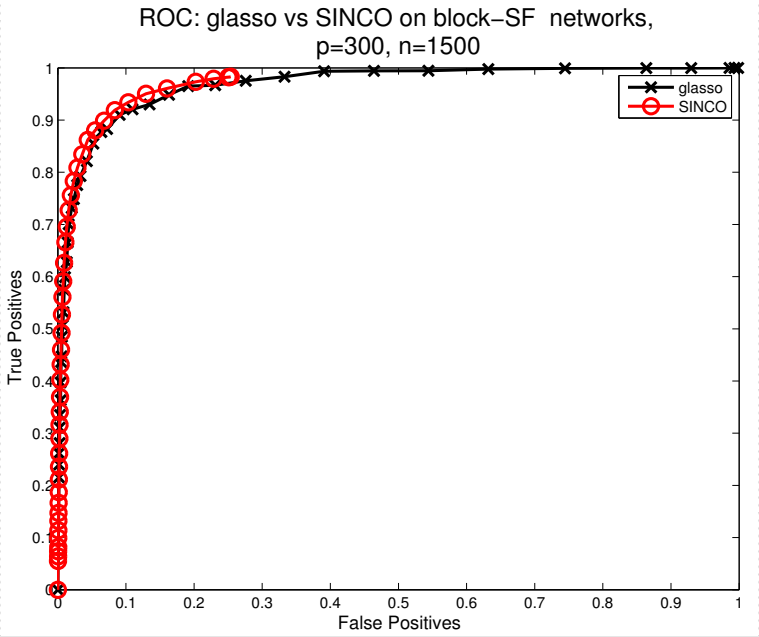




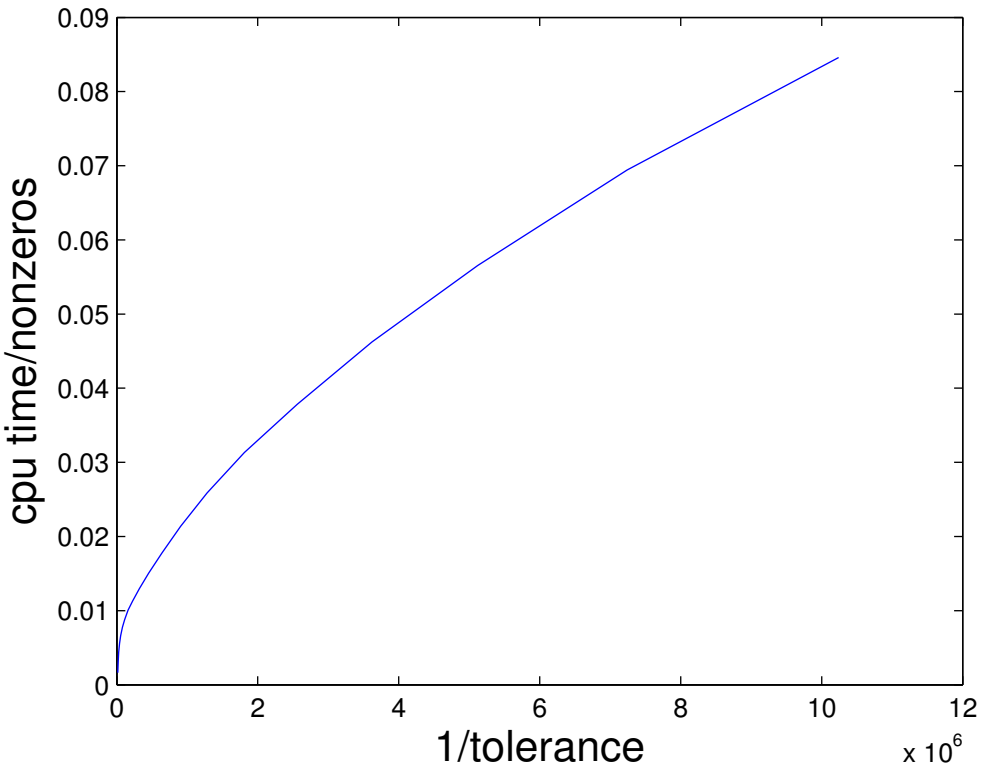


CPU time scaling with p, with proportional # of nonzeros, scale-free networks





What about the CPU time scaling with tol?



Conclusions and Future work

- We have developed a very simple coordinate ascent method for the primal sparse inverse covariance selection problem.
- The method has a greedy nature which allows it to produce solutions that seem to follow the regularization path.
- It is possible to apply this method until desired sparsity is achieved.
- The method is theoretically convergent but can we show any convergence rate?
- Many improvements are possible.
- The method can be adapted to incremental mode and lends itself well to parallelization.

SINCO code is in C++ with Matlab interface and is open source, available from www.coin-or.org

Thank you!