Sparse Inverse Covariance Estimation Using Quadratic Approximation

Inderjit S. Dhillon
Dept of Computer Science
UT Austin

IPAM
Los Angeles, California

Jan 17, 2013

Joint work with C. Hsieh, M. Sustik and P. Ravikumar

Inverse Covariance Estimation

- Given: n i.i.d. samples $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$, $\mathbf{y_i} \sim \mathcal{N}(\mu, \Sigma)$,
- Goal: Estimate the inverse covariance $\Theta = \Sigma^{-1}$.
- The sample mean and covariance are defined by

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y_i} \quad \text{and} \quad S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y_i} - \hat{\mu})(\mathbf{y_i} - \hat{\mu})^T.$$

- Given the n samples, the likelihood is

$$\begin{aligned} P(\mathbf{y_1}, \ldots, \mathbf{y_n}; \hat{\mu}, \Theta) &\propto \prod_{i=1}^{n} (\det \Theta)^{1/2} \exp\left(-\frac{1}{2}(\mathbf{y_i} - \hat{\mu})^T \Theta (\mathbf{y_i} - \hat{\mu})\right) \\ &= (\det \Theta)^{n/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{y_i} - \hat{\mu})^T \Theta (\mathbf{y_i} - \hat{\mu})\right). \end{aligned}$$

Inverse Covariance Estimation

- The log likelihood can be written as

$$\log(P(\mathbf{y_1}, \ldots, \mathbf{y_n}; \hat{\mu}, \Theta)) = \frac{n}{2}\log(\det\Theta) - \frac{n}{2}\mathrm{tr}(\Theta S) + \text{constant}.$$

- The maximum likelihood estimator of Θ is

$$\Theta = \arg\min_{X \succ 0}\{-\log\det X + \mathrm{tr}(SX)\}.$$

- In high-dimensions $(p > n)$, the sample covariance matrix S is singular.

Structure for Gaussian Markov Random Field

- The nonzero pattern of Θ is important.
- Conditional independence is reflected as zeros in Θ:

  $\Theta_{ij} = 0 \Leftrightarrow$ y_i and y_j are conditionally independent given other variables.

- Each Gaussian distribution can be represented by a pairwise Gaussian Markov Random Field (GMRF).
- In a GMRF $G = (V, E)$, each node corresponds to a variable, and each edge corresponds to a non-zero entry in Θ.
- In many cases of interest, Θ is sparse.

L1-regularized covariance selection

- A sparse inverse covariance matrix is preferred – add ℓ_1 regularization to promote sparsity.
- The resulting optimization problem:

$$\Theta = \arg\min_{X \succ 0} \left\{ -\log\det X + \operatorname{tr}(SX) + \lambda \|X\|_1 \right\} = \arg\min_{X \succ 0} f(X),$$

  where $\|X\|_1 = \sum_{i,j=1}^{n} |X_{ij}|$.
- Regularization parameter $\lambda > 0$ controls the sparsity.
- Can be extended to a more general regularization term: $\|\Lambda \circ X\|_1 = \sum_{i,j=1}^{n} \lambda_{ij} |X_{ij}|$

Prior Work

- COVSEL: Block coordinate descent method with interior point solver for each block (Banerjee, El Ghaoui & d'Aspremont, 2007).
- GLASSO : Block coordinate descent method with coordinate descent solver for each block (Friedman, Hastie & Tibshirani, 2007).
- VSM: Nesterov's algorithm (Lu, 2009).
- PSM : Projected Subgradient Method (Duchi, Gould & Koller, 2008).
- SINCO : Greedy coordinate descent method (Scheinberg & Rish, 2009).
- ALM : Alternating Linearization Method (Scheinberg, Ma & Goldfarb, 2010).
- IPM : Inexact interior point method (Li & Toh, 2010).
- PQN : Projected Quasi-Newton to solve dual (Schmidt et al, 2009).

Second Order Method

- Newton method for twice differentiable function:

$$\mathbf{x} \leftarrow \mathbf{x} - \eta(\nabla^2 f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$$

- However, the sparse inverse covariance estimation objective

$$f(X) = -\log\det X + \operatorname{tr}(SX) + \lambda\|X\|_1$$

  is not differentiable.

- Most current solvers are first-order methods:

  Block Coordinate Descent (GLASSO), projected gradient descent (PSM), greedy coordinate descent (SINCO), alternating linearization method (ALM).

Quadratic Approximation

- Write objective as $f(X) = g(X) + h(X)$, where

$$g(X) = -\log\det X + \operatorname{tr}(SX) \text{ and } h(X) = \lambda\|X\|_1.$$

- $g(X)$ is twice differentiable while $h(X)$ is convex but non-differentiable — we can only form quadratic approximation for $g(X)$.
- The quadratic approximation of $g(X_t + \Delta)$ is

$$\bar{g}_{X_t}(\Delta) = \operatorname{tr}((S - W_t)\Delta) + (1/2)\operatorname{tr}(W_t\Delta W_t\Delta) - \log\det X_t + \operatorname{tr}(SX_t),$$

where $W_t = (X_t)^{-1}$.
- Note that

$$\operatorname{tr}(W_t\Delta W_t\Delta) = \operatorname{vec}(\Delta)^T(W_t \otimes W_t)\operatorname{vec}(\Delta)$$

Descent Direction

- Define the generalized Newton direction:

$$D = \arg\min_{\Delta} \bar{g}_{X_t}(\Delta) + \lambda\|X + \Delta\|_1,$$

where $\bar{g}_{X_t}(\Delta) \equiv g(X_t + \Delta) = \operatorname{tr}((S - W_t)\Delta) + \frac{1}{2}\operatorname{tr}(W_t \Delta W_t \Delta)$.

- Can be rewritten as a Lasso type problem with $p(p+1)/2$ variables:

$$\frac{1}{2}\operatorname{vec}(\Delta)^T (W_t \otimes W_t) \operatorname{vec}(\Delta) + \operatorname{vec}(S - W_t)^T \operatorname{vec}(\Delta) + \lambda\|\operatorname{vec}(\Delta)\|_1.$$

- Coordinate descent method is efficient at solving Lasso type problems.

Coordinate Descent Updates

- Can use cyclic coordinate descent to solve $\arg\min_{\Delta}\{\bar{g}_{X_t}(\Delta) + \lambda\|\Delta\|_1\}$:
    - Generate a sequence $D_1, D_2 \ldots$, where D_i is updated from D_{i-1} by only changing one variable.
    - Variables are selected in cyclic order.
- Naive approach has an update cost of $O(p^2)$ because

$$\nabla_i \bar{g}(\Delta) = ((W_t \otimes W_t)\,\mathrm{vec}(\Delta) + \mathrm{vec}(S - W_t))_i$$

- Key point 1: we can reduce the cost from $O(p^2)$ to $O(p)$.

Coordinate Descent Updates

- Each coordinate descent update:

$$\bar{\mu} = \arg\min_{\mu} \bar{g}(D + \mu(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T)) + 2\lambda|X_{ij} + D_{ij} + \mu|$$

$$D_{ij} \leftarrow D_{ij} + \bar{\mu}$$

- The one-variable problem can be simplified as

$$\frac{1}{2}(W_{ij}^2 + W_{ii}W_{jj})\mu^2 + (S_{ij} - W_{ij} + \mathbf{w}_i^T D\mathbf{w}_j)\mu + \lambda|X_{ij} + D_{ij} + \mu|$$

- Quadratic form with L1 regularization — soft thresholding gives the exact solution.

Efficient solution of one-variable problem

- If we introduce $a = W_{ij}^2 + W_{ii}W_{jj}$, $b = S_{ij} - W_{ij} + \mathbf{w}_i^T D\mathbf{w}_j$, and $c = X_{ij} + D_{ij}$, then the minimum is achieved for:

$$\mu = -c + \mathcal{S}(c - b/a, \lambda/a),$$

  where $\mathcal{S}(z, r) = \text{sign}(z)\max\{|z| - r, 0\}$ is the soft-thresholding function.
- The main cost arises while computing $\mathbf{w}_i^T D\mathbf{w}_j$: direct computation requires $O(p^2)$ flops.
- Instead, we maintain $U = DW$ after each coordinate updates, and then compute $\mathbf{w}_i^T\mathbf{u}_j$ — only $O(p)$ flops per updates.

Line Search

- Adopt Armijo rule: try step-sizes $\alpha \in \{\beta^0, \beta^1, \beta^2, \ldots\}$ st $X_t + \alpha D_t$:
  1. is positive definite
  2. satisfies a sufficient decrease condition

$$f(X_t + \alpha D_t) \quad \leq \quad f(X_t) + \alpha \sigma \Delta_t$$

where $\Delta_t = \text{tr}(\nabla g(X_t) D_t) + \lambda \|X_t + D_t\|_1 - \lambda \|X_t\|_1$.

- Both conditions can be checked by performing Cholesky factorization — $O(p^3)$ flops per line search iteration.
  - Can possibly do better by using Lanczos [K.C.Toh]

Free and Fixed Set — Motivation

- Recall the time cost for finding descent direction:

  $O(p^2)$ variables, each update needs $O(p)$ flops $\rightarrow$ total $O(p^3)$ flops per sweep.
- Our goal: Reduce the number of variable updates from $O(p^2)$ to $\|X_t\|_0$.
- $\|X_t\|_0$ can be much smaller than $O(p^2)$ as the suitable λ should give a sparse solution.
- Our strategy: before solving the Newton direction, "guess" the variables that should be updated.

Free and Fixed Sets

- $(X_t)_{ij}$ belongs to *fixed* set if and only if

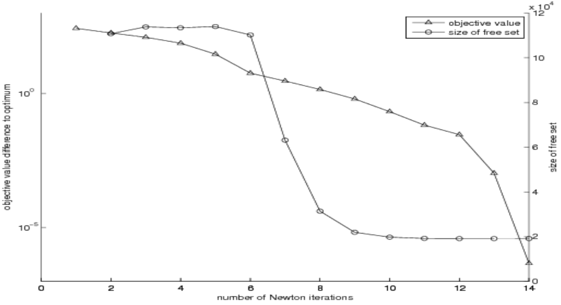
$$|\nabla_{ij} g(X_t)| < \lambda, \text{ and } (X_t)_{ij} = 0.$$

- The remaining variables constitute the *free* set.
- We then perform the coordinate descent updates only on *free* set.

Size of *free* set

- In practice, the size of *free* set is small.
- Take Hereditary dataset as an example:

  $p = 1869$, number of variables $= p^2 = 3.49$ million. The size of *free* set drops to $20,000$ at the end.



Block-diagonal Structure

- Recently, (Mazumdar and Hastie, 2012) and (Witten et al, 2011) proposed a block decomposition approach.
- Consider the thresholded covariance matrix $E_{ij} = \max(|S_{ij}| - \lambda, 0)$.
- When E is block-diagonal, the solution is also block-diagonal:

$$E = \begin{bmatrix} E_1 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & E_n, \end{bmatrix}, \quad \Theta^* = \begin{bmatrix} \Theta_1^* & 0 & \dots & 0 \\ 0 & \Theta_2^* & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \Theta_n^*, \end{bmatrix}$$

- Based on this approach, the original problem can be decomposed into n sub-problems.

Block-diagonal Structure for “Free”

- Our method automatically discovers the block-diagonal structure too.
- Key observation: off-diagonal blocks are always in the *fixed* set.
- Recall the definition of fixed set: $|\nabla_{ij} g(X_t)| < \lambda$ and $(X_t)_{ij} = 0$.
- For (i,j) in off-diagonal blocks:

  1. Initialize X to be a diagonal matrix, i.e., $(X_0)_{ij} = 0$.
  2. $\nabla_{ij} g(X_t) = S_{ij} - (X_t^{-1})_{ij} = S_{ij}$.
  3. $E_{ij} = \max(|S_{ij}| - \lambda, 0) = 0$ implies $|\nabla_{ij} g(X_t)| < \lambda$. So (i,j) is always in the fixed set.

- Off-diagonal blocks are always 0, so QUIC gets the speedup for free.

Final Algorithm

QUIC: QUadratic approximation for sparse Inverse Covariance estimation

Input: Empirical covariance matrix S, scalar λ, initial X_0.

For $t = 0, 1, \ldots$

1. Compute $W_t = X_t^{-1}$.
2. Form the second order approximation $\bar{g}_{X_t}(X)$ to $g(X)$ around X_t.
3. Partition variables into free and fixed sets
4. Use coordinate descent to find descent direction:
   $D_t = \arg\min_{\Delta} \bar{f}_{X_t}(X_t + \Delta)$ over set of free variables, (A *Lasso* problem.)
5. Use an *Armijo*-rule based step-size selection to get α s.t.
   $X_{t+1} = X_t + \alpha D_t$ is positive definite and objective sufficiently decreases.

Methods included in comparisons

- QUIC: Proposed method.
- ALM : Alternating Linearization Method (Scheinberg et al, 2010).
- GLASSO : Block coordinate descent (Friedman et al, 2007).
- PSM : Projected Subgradient Method (Duchi, Gould & Koller, 2008).
- SINCO : Greedy coordinate descent (Scheinberg & Rish, 2009).
- IPM : Inexact interior point method (Li and Toh, 2010).

Synthetic datasets

We generate the two following types of graph structures for GMRF:

- Chain graphs: The ground truth inverse covariance matrix Σ^{-1} is set to be $\Sigma^{-1}_{i,i-1} = -0.5$ and $\Sigma^{-1}_{i,i} = 1.25$.
- Graphs with Random Sparsity Structures:
    - First, generate a sparse matrix U with nonzero elements equal to ± 1,
    - Set Σ^{-1} to be $U^T U$
    - Add a diagonal term to ensure Σ^{-1} is positive definite.

  Control the number of nonzeros in U so that the resulting Σ^{-1} has approximately $10p$ nonzero elements.

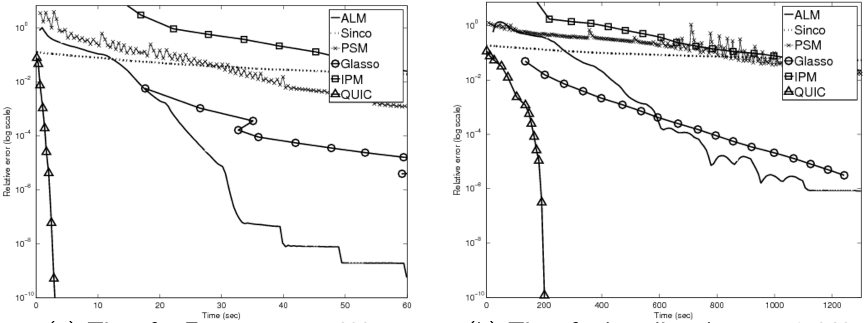
Experimental settings

- Test under two values of λ: one discovers correct number of nonzeros, and one discovers 5 times the number of nonzeros.
- For each distribution we draw $n = p/2$ i.i.d. samples as input.
- We report the time for each algorithm to achieve ϵ-accurate solution: $f(X_t) - f(X^*) < \epsilon f(X^*)$.
- * indicates the run time exceeded 30,000 seconds (8.3 hours).

Results for Synthetic datasets

Dataset setting				Time (in seconds)					
pattern	p	λ	ϵ	QUIC	ALM	Glasso	PSM	IPM	Sinco
chain	1000	0.4	10^{-2}	**0.30**	18.89	23.28	15.59	86.32	120.0
			10^{-6}	**2.26**	41.85	45.1	34.91	151.2	520.8
chain	10000	0.4	10^{-2}	**216.7**	13820	*	8450	*	*
			10^{-6}	**986.6**	28190	*	19251	*	*
random	1000	0.12	10^{-2}	**0.52**	42.34	10.31	20.16	71.62	60.75
			10^{-6}	**1.2**	28250	20.43	59.89	116.7	683.3
		0.075	10^{-2}	**1.17**	65.64	17.96	23.53	78.27	576.0
			10^{-6}	**6.87**	*	60.61	91.7	145.8	4449
random	10000	0.08	10^{-2}	**337.7**	26270	21298	*	*	*
			10^{-6}	**1125**	*	*	*	*	*
		0.04	10^{-2}	**803.5**	*	*	*	*	*
			10^{-6}	**2951**	*	*	*	*	*

Real datasets



(a) Time for Estrogen, $p = 692$

(b) Time for hereditarybc, $p = 1,869$

Figure: Comparison of algorithms on real datasets. The results show QUIC converges faster than other methods.

Divide-and-conquer QUIC – Motivation

- All solvers require at least $O(p^3)$ time per iteration for computing X^{-1} (the gradient of $\log\det X$).
- Hard to scale to problems with > 10000 variables.
- For example, QUIC takes more than 10 hours on a climate dataset with 10,512 variables.
- Further enhancement: a divide-and-conquer procedure:
  - Divide the problem into smaller subproblems.
  - Subproblems can be solved much faster.

The divide-and-conquer framework

Input: Empirical covariance matrix S, scalar λ

Output: The solution Θ^*

- Obtain a partition of the nodes $\{\mathcal{V}_c\}_{c=1}^{k}$
- **for** $c = 1, 2, \ldots, k$ **do**

    Solve the subproblem of variables in $\mathcal{V}_c$ to get $\Theta^{(c)}$

- Form the block-diagonal estimate $\bar{\Theta}$ by

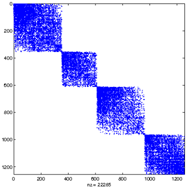
$$\bar{\Theta} = \begin{bmatrix} \Theta^{(1)} & 0 & \ldots & 0 \\ 0 & \Theta^{(2)} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \Theta^{(k)} \end{bmatrix}.$$

- Use $\bar{\Theta}$ as an initial point to solve the whole problem.

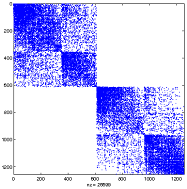
The divide-and-conquer framework

- Final stage (solving the whole problem) is efficient if $\|\Theta^* - \bar{\Theta}\|$ is small.
- The partition has to be balanced:

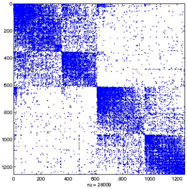
  Time for solving subproblems is $O(k(p/k)^3) = O(p^3/k^2)$ if the partition is balanced.
- Can apply the divide-and-conquer procedure Hierarchically
  - For solving subproblems, we can again apply divide-and-conquer.
  - Initial time can be further reduced.



The recovered $\bar{\Theta}$ from level-2 clusters



The recovered $\bar{\Theta}$ from level-1 clusters



The inverse covariance matrix Θ^*

Bounding the distance between Θ^* and $\bar{\Theta}$

- The divide-and-conquer procedure is efficient if $\|\bar{\Theta} - \Theta^*\|$ is small.
- Recently, (Mazumdar and Hastie, 2012) and (Witten et al, 2011) showed that $\Theta^* = \bar{\Theta}$ when $|S_{ij}| \le \lambda$ for all i, j in different blocks.
- However, in most real examples, a perfect partitioning does not exist.
- We derive a bound on $\|\Theta^* - \bar{\Theta}\|$: defining

$$E_{ij} = \begin{cases} 0 & \text{if } i,j \text{ are in the same cluster,} \\ \max(|S_{ij}| - \lambda, 0) & \text{otherwise.} \end{cases}$$

- **Theorem 1**: If $\exists \gamma > 0$ such that $\|E\|_2 \le (1-\gamma)\frac{1}{\|\bar{W}\|_2}$, then

$$\|\Theta^* - \bar{\Theta}\|_F \le \frac{p \max(\sigma_{max}(\bar{\Theta}), \sigma_{max}(\Theta^*))^2 \sigma_{max}(\bar{\Theta})}{\gamma \min(\sigma_{min}(\Theta^*), \sigma_{min}(\bar{\Theta}))} \|E\|_F,$$

where $\sigma_{min}(\cdot), \sigma_{max}(\cdot)$ denote the minimum/maximum singular values.

The clustering algorithm

- Based on Theorem 1, minimizing $\|E\|_F$ can be cast as a relaxation of the problem of minimizing $\|\bar{\Theta} - \Theta^*\|_F$.
- To minimize $\|E\|_F$, we want to find a partition such that the sum of off-diagonal block entries of S^λ is minimized, where

$$(S^\lambda)_{ij} = \max(|S_{ij}| - \lambda, 0)^2 \; \forall \; i \neq j \quad \text{and} \quad S^\lambda_{ij} = 0 \; \forall i = j.$$

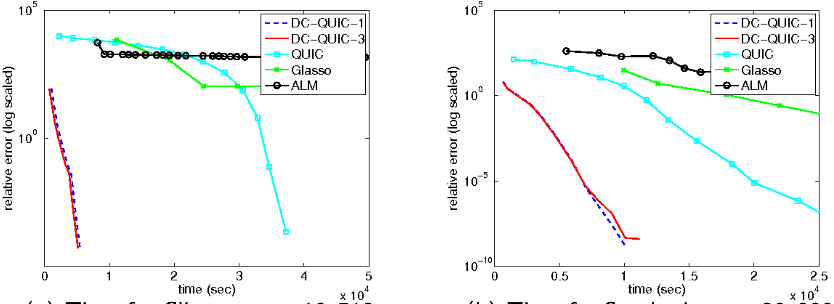
- Therefore, the clusters can be identified by running spectral clustering algorithms on S^λ.
- Time needed for clustering is $O(kp^2)$

  less than the time cost for one iteration $O(p^3)$
- Can speed up by using the Graclus multilevel algorithm, which is a faster heuristic to minimize normalized cut.

Methods in our comparisons

- DC-QUIC-1: Divide-and-Conquer QUIC with 1 level clustering.
- DC-QUIC-3: Divide-and-Conquer QUIC with 3 levels of hierarchical clustering.
- QUIC: Original QUIC (Hsieh, Sustik, Dhillon & Ravikumar, 2011).
- Glasso: The block coordinate descent algorithm proposed in (Friedman, Hastie & Tibshirani, 2008).
- ALM: The alternating linearization algorithm proposed by (Scheinberg, Ma & Goldfarb, 2010).

Performance on real datasets



(a) Time for Climate, $p = 10,512$

(b) Time for Synthetic, $p = 20,000$

Figure: Comparison of algorithms on real datasets. The results show DC-QUIC converges faster than other methods.

Conclusions

- Proposed a quadratic approximation method for sparse inverse covariance learning (QUIC).
- Three key ingredients:
    - Exploit structure of Hessian
        - we have done this in the context of coordinate descent
        - Nocedal & colleagues(2012) have recently developed other methods to exploit structure of Hessian, e.g., Newton-CG
    - Armijo-type stepsize rule
    - Division into *free* and *fixed* sets
- Initial paper published in NIPS 2011:
    - "Sparse Inverse Covariance Matrix Estimation using Quadratic Approximation", NIPS, 2011.
- Divide-and-conquer QUIC published in NIPS 2012:
    - "A Divide-and-Conquer Procedure for Sparse Inverse Covariance Estimation", NIPS, 2012.
- Journal version coming soon......

References

[1] C. J. Hsieh, M. Sustik, I. S. Dhillon, and P. Ravikumar. *Sparse Inverse Covariance Matrix Estimation using Quadratic Approximation*. NIPS, 2011.
[2] C. J. Hsieh, I. S. Dhillon, P. Ravikumar, A. Banerjee. *A Divide-and-Conquer Procedure for Sparse Inverse Covariance Estimation*. NIPS, 2012.
[3] P. A. Olsen, F. Oztoprak, J. Nocedal, and S. J. Rennie. *Newton-Like Methods for Sparse Inverse Covariance Estimation*. Optimization Online, 2012.
[4] O. Banerjee, L. El Ghaoui, and A. d'Aspremont *Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data*. JMLR, 2008.
[5] J. Friedman, T. Hastie, and R. Tibshirani. *Sparse inverse covariance estimation with the graphical lasso*. Biostatistics, 2008.
[6] J. Duchi, S. Gould, and D. Koller. *Projected subgradient methods for learning sparse Gaussians*. UAI, 2008.
[7] L. Li and K.-C. Toh. *An inexact interior point method for l1-reguarlized sparse covariance selection*. Mathematical Programming Computation, 2010.
[8] K. Scheinberg, S. Ma, and D. Glodfarb. *Sparse inverse covariance selection via alternating linearization methods*. NIPS, 2010.
[9] K. Scheinberg and I. Rish. *Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach*. Machine Learning and Knowledge Discovery in Databases, 2010.
[10] Z. Lu. *Smooth optimization approach for sparse covariance selection*. SIAM J. Optim, 2009.
[11] M. Schmidt, E. van den Berg, M. Friedlander, and K. Murphy. *Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm* . AISTATS, 2009.