Computation Proof for Stat771 Project

Yuan Li

May 5th, 2015

1. Simulation Setting

Following is a small simulation study to show how the algorithms in the paper will work. This simulation is similar with that in the paper, but here I use different levels for the values of p, and run the simulation on a different machine (Nebula-9 on biostat server). In this simulation I generate data with 20 independent observations distributed as $N(0, \Sigma)$, where Σ is a $p \times p$ matrix that takes one of three forms: (1) $\Sigma = \mathbf{I}$, (2) Σ is a block diagonal matrix with a single $(p/2) \times (p/2)$ block with all off-diagonal elements equal to 0.5, and all diagonal elements of Σ equal 1, and (3) all off-diagonal elements of Σ equal 0.5 and all diagonal elements equal 1. I use three different values of $p : p = 100, 800, 1500$. I choose λ in each simulation in order to achieve a desired "sparsity level", measured by the fraction of completely unconnected nodes in the graphical lasso solution. The three sparsity levels are 0.2,0.5,0.9. The results for the simulation with three different sparsity levels are as following:

2. Simulation Results

```r
### Sparsity Level=0.2
noquote(Sparsity2)
```

```
## $`p=100`
##             Simu_1       Simu_2       Simu_3
## Original    0.016(0.001) 0.019(0.001) 0.018(0)
## Algorithm_1 0.01(0)      0.015(0.001) 0.014(0.001)
## Algorithm_2 0.007(0.001) 0.019(0.002) 0.022(0.001)
## MB_Approx   0.007(0)     0.007(0)     0.006(0)
##
## $`p=800`
##             Simu_1        Simu_2        Simu_3
## Original    19.012(1.166) 29.912(2.133) 25.294(1.683)
## Algorithm_1 12.753(1.031) 20.01(1.538)  13.714(1.175)
## Algorithm_2 4.94(0.694)   15.175(0.853) 19.545(0.86)
## MB_Approx   6.904(0.206)  6.611(0.465)  7.061(0.64)
##
## $`p=1500`
##             Simu_1         Simu_2         Simu_3
## Original    141.521(10.875) 208.372(8.897) 158.607(8.109)
## Algorithm_1 83.205(2.607)   145.959(9.734) 111.343(6.995)
## Algorithm_2 47.415(1.986)   108.29(9.194)  129.412(3.894)
## MB_Approx   46.619(0.984)   41.587(2.398)  46.583(2.682)
```

```r
### Sparsity Level=0.5
noquote(Sparsity5)
```

```
## $`p=100`
```

```
##              Simu_1   Simu_2       Simu_3
## Original    0.012(0) 0.015(0.001) 0.013(0)
## Algorithm_1 0.006(0) 0.007(0)     0.006(0)
## Algorithm_2 0.004(0) 0.01(0.001)  0.007(0)
## MB_Approx   0.006(0) 0.007(0.001) 0.006(0)
##
## $`p=800`
##              Simu_1       Simu_2       Simu_3
## Original    5.354(0.109) 8.294(0.544) 6.816(0.259)
## Algorithm_1 2.782(0.3)   3.381(0.38)  2.269(0.143)
## Algorithm_2 1.48(0.135)  3.499(0.406) 2.826(0.123)
## MB_Approx   5.742(0.277) 5.931(0.437) 6(0.419)
##
## $`p=1500`
##              Simu_1        Simu_2         Simu_3
## Original    59.407(1.56)  104.072(4.133) 89.714(3.758)
## Algorithm_1 27.856(0.424) 34.803(1.611)  27.087(0.921)
## Algorithm_2 11.081(0.264) 33.06(2.699)   28.25(1.216)
## MB_Approx   41.891(1.634) 39.66(2.217)   39.856(1.761)
```

```r
### Sparsity Level=0.9
noquote(Sparsity9)
```

```
## $`p=100`
##              Simu_1   Simu_2       Simu_3
## Original    0.009(0) 0.011(0.001) 0.009(0)
## Algorithm_1 0.003(0) 0.003(0)     0.003(0)
## Algorithm_2 0.004(0) 0.004(0)     0.004(0)
## MB_Approx   0.006(0) 0.006(0)     0.005(0)
##
## $`p=800`
##              Simu_1       Simu_2       Simu_3
## Original    4.236(0.264) 5.478(0.107) 5.14(0.39)
## Algorithm_1 1.531(0.158) 0.863(0.008) 0.913(0.061)
## Algorithm_2 1.317(0.066) 1.082(0.049) 1.047(0.042)
## MB_Approx   6.135(0.669) 6.303(0.641) 4.96(0.229)
##
## $`p=1500`
##              Simu_1        Simu_2        Simu_3
## Original    48.243(1.711) 65.071(2.104) 61.726(2.899)
## Algorithm_1 10.277(0.387) 8.237(0.22)   7.895(0.263)
## Algorithm_2 9.373(0.281)  10.481(0.449) 9.324(0.265)
## MB_Approx   42.054(2.333) 39.111(1.478) 36.184(2.601)
```

3. Discussion

First we can see that the speed of Algorithm 1 and Algorithm 2 in this paper is much faster than the original standard graphical lasso algorithm. Moreover we can see that Algorithm 2 is faster than Algorithm 1 in Simulation 1 since the true Σ is diagonal the first step in Algorithm 1 is unnecessary. On the other hand we can see that in Simulation 3 Algorithm 1 performs better than Algorithm 2 since in this case the true Σ is dense so the connected nodes in the graphical lasso solution mostly belong to a single connected component rather than multiple connected components. These conclusions are consistent with that in the paper.