Experiment

Rui Zhang

February 5, 2018

1 Chi-square distribution of $T_N(r)$

1.1 Experiment Setting

- N: sample size
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1 \times n_2}$: true value. $Y^* = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-\theta, \theta]$.
- Δ_{ij}: population drift.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0, \sigma^2)$, we don't need a large N to guarantee the convergence.
- Ω : set of observed positions.
- M_{ij}: observed values. $M = P_\Omega(Y^* + N^{-1/2}\Delta + \varepsilon)$.
- $T_N(r)$: test statistics. $T_N(r) := N \min_{Y \in \mathcal{M}_r} \sum_{(i,j)\in\Omega} w_{ij}(Y_{ij} - M_{ij})^2$
- w: weight. $w_{ij} = 1/\sigma^2$
- δ_r: theoretical noncentrality parameter. $\delta_r = \min_{H \in P_\Omega(\mathcal{T}_{\mathcal{M}_r})} \sum_{(i,j)\in\Omega} \sigma_{ij}^{-2}(\Delta_{ij} - H_{ij})^2$
- In this experiment, $n_1 = 40,\ n_2 = 50, rank(Y^*) = 11, |\Omega| = 1000$ and the generic bound is 12

For one N, $Y*$, Δ and Ω, multiple $\varepsilon' s$ are generated (say, N_{rep} times). Then, I solved the least square problem and get $T_N(r)$, N_{rep} times. Qqlot all N_{rep} $T_N(r)'s$ with corresponding chi-squares quantiles. In this experiment $N_{rep} = 200$.

In the experiment I use soft-thresholded svd to solve the least square problem [1]. This method use the same objective function as our's.

When solving the least square problem, an optimal solution is never obtained (we stop the iteration when the number of iterations reaches some large value or the change of objective function is less than some value). Therefore the $T_N(r)$ we get contains three parts: central chi-square induced by $N^{1/2}\varepsilon$, noncentrality induced by Δ and the noncentrality induced by error of the optimization methods ($|Y_{opt} - \hat{Y}|$, where $Y_{opt} = arg \min_{Y} \sum_{(i,j)\in\Omega} (Y_{ij} - M_{ij})^2$ and $\hat{Y}$ is the approximate solution given by the optimization methods).

When the first two parts dominate, the results are just as the theoretical results proved by Prof. Shapiro. When the third part dominates, the effect of optimization error acts like adding a noncentrality parameter (which I estimate by $\delta_r^{opt} = N \min_{rank(Y)=r} \sum_{(i,j)\in\Omega} (Y^*_{ij} - Y_{ij})^2$).

I will try more methods mentioned in this website(`http://perception.csl.illinois.edu/matrix-rank/sample_code.html`).

1.2 Results

I will talk about the result in two scenarios, converged case and diverged case. *Both these real matrix Y^*'s satisfy the sufficient condition that column vectors $w_j^T \otimes v_i, (i,j) \in \Omega^c$ are linearly independent. (Of course, they also satisfy the necessary condition that at least r entries are observed in each column and row.)*

1.2.1 Converged case

- Central case: $\Delta = 0, \theta = 20, \sigma = 5,\ max\ iteration = 5 * 10^4$

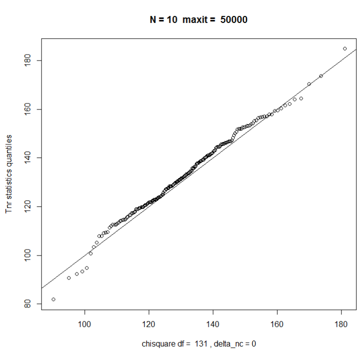

Figure 1: X axis is the quantiles of central chi-square with degree of freedom 131

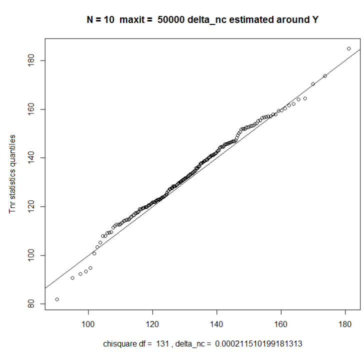

Figure 2: X axis is the quantiles of noncentral chi-square with $df_r = 131$ and noncentrality parameter δ_r^{opt}

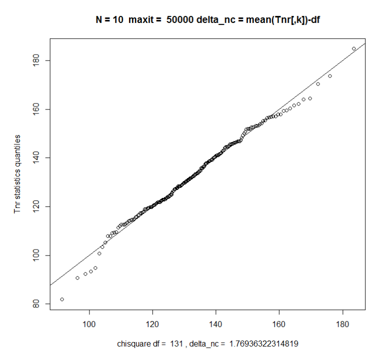

Figure 3: Here, the noncentrality parameter is estimated by the sample mean of $T_N(r) - df_r$

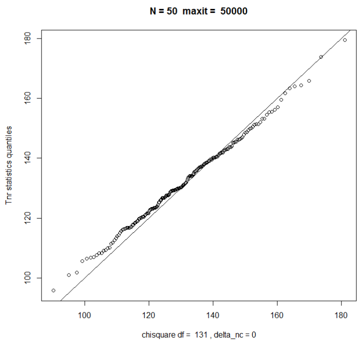

Figure 4: $T_N(r)$ is less dispersed than the chi-sqaure distribution, i.e. light tailed

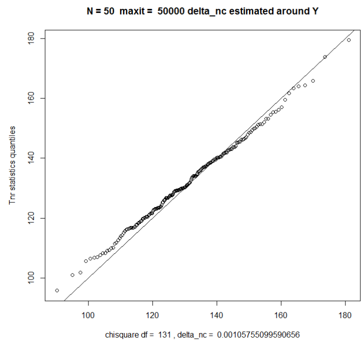

Figure 5

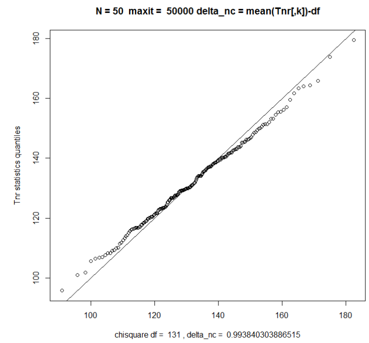

Figure 6

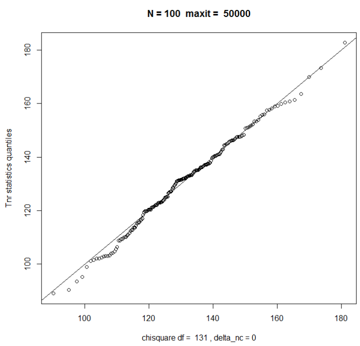

Figure 7: a little bit left skew

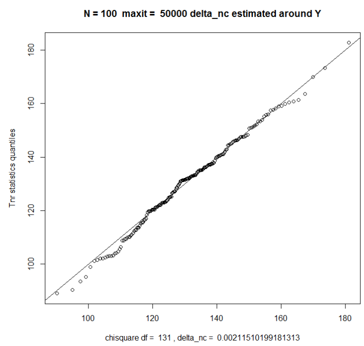

Figure 8

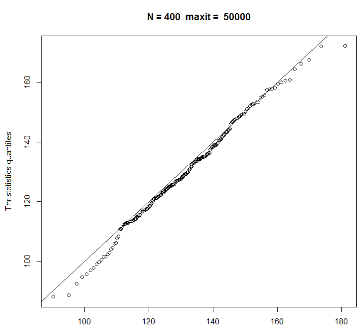

Figure 9

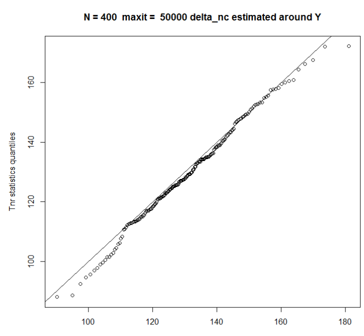

Figure 10

- Non Central case: $\Delta = 4, \theta = 20, \sigma = 5,\ max\ iteration = 10^4$

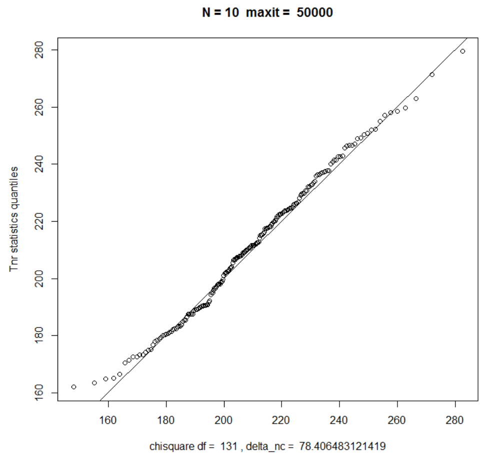

Figure 11: Noncentral chi-square fits $T_N(r)$ well

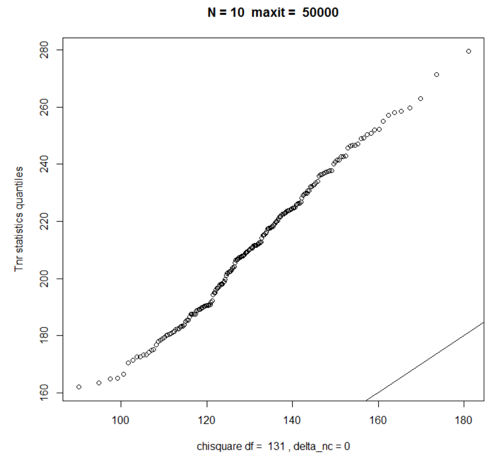

Figure 12

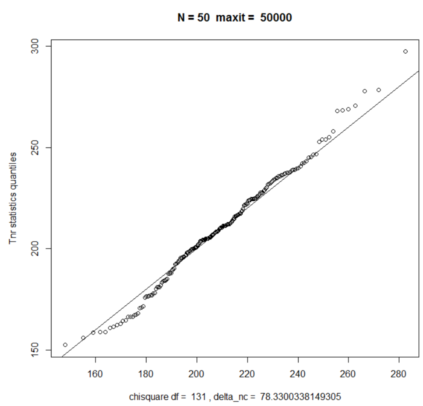

Figure 13

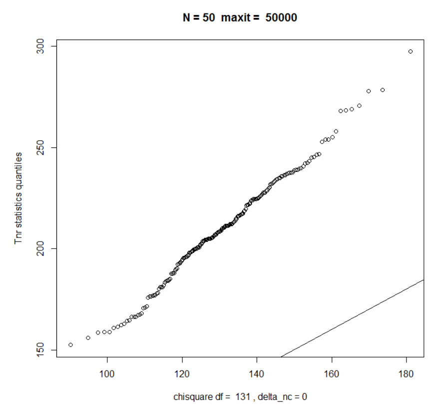

Figure 14

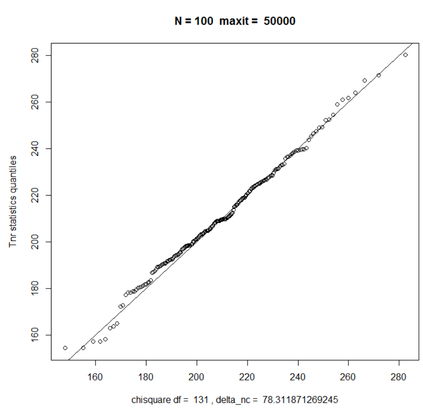

Figure 15

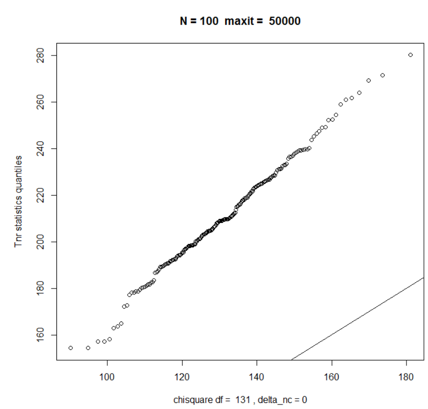

Figure 16

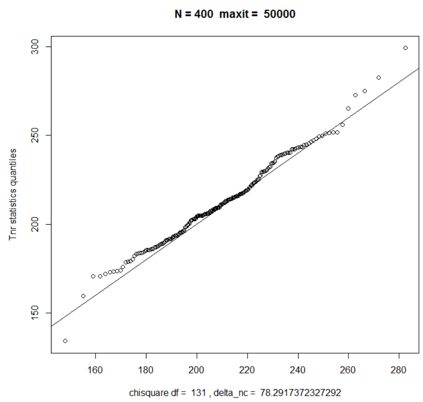

Figure 17

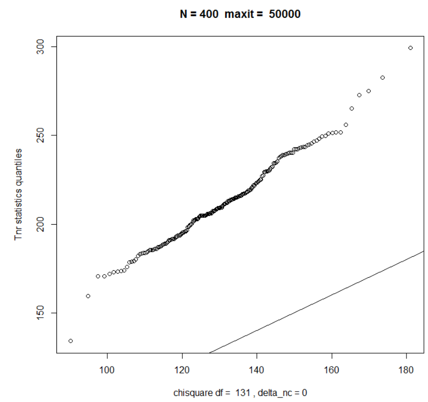

Figure 18

1.2.2 Diverged case

In this part, I try to show the effect of the optimization error. Since in my experiments $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$, large N has no help for the asymptotic normality, instead it enlarges the effect of optimization error. It seems the error increases as the sigma increases.

- Central case1: $\Delta = 0, \theta = 10, \sigma = 1,$ $max\ iteration = 10^5$

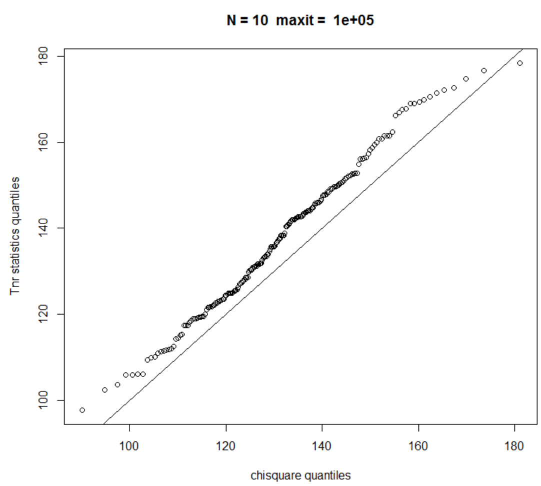

Figure 19

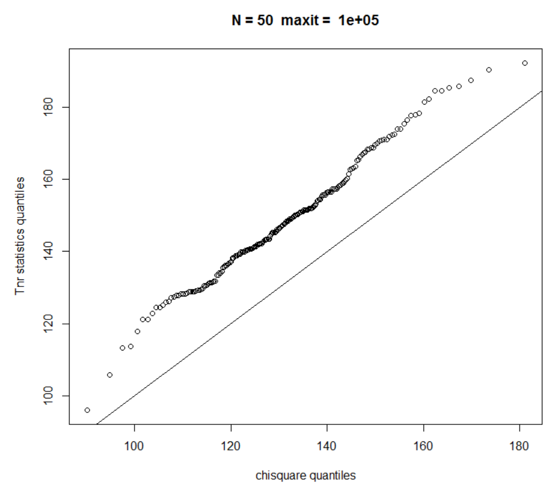

Figure 20: The gap become larger as N increases

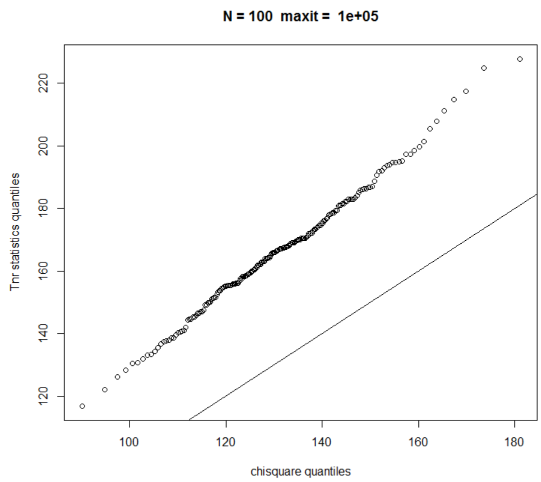

Figure 21

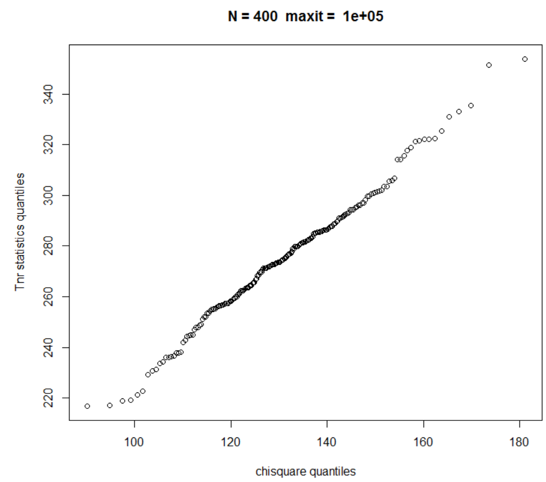

Figure 22

- Central case2: $\Delta = 0, \theta = 20, \sigma = 5,$ $max\ iteration = 10^5$. This part I try to show, the effect of optimization error can be estimate by the δ_r^{opt}.

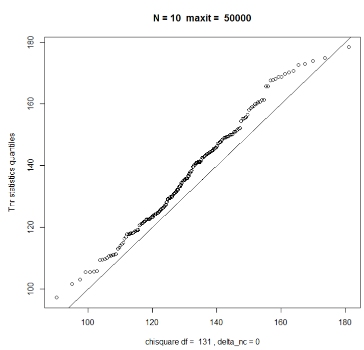

Figure 23

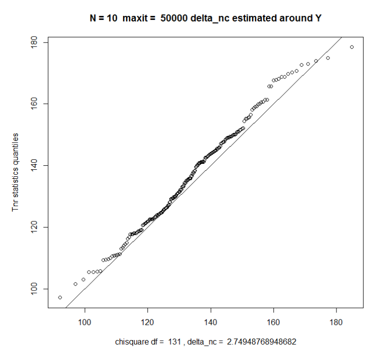

Figure 24

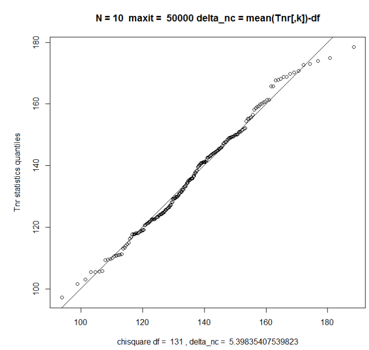

Figure 25

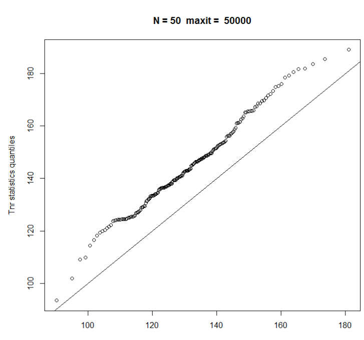

Figure 26

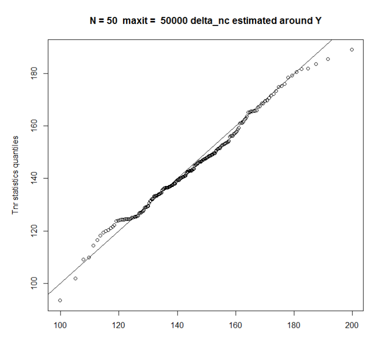

Figure 27

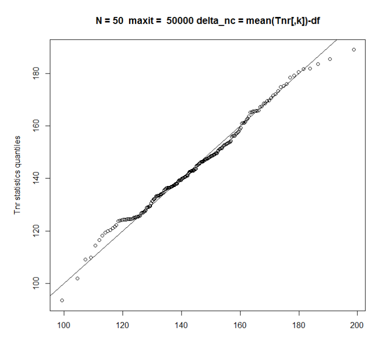

Figure 28

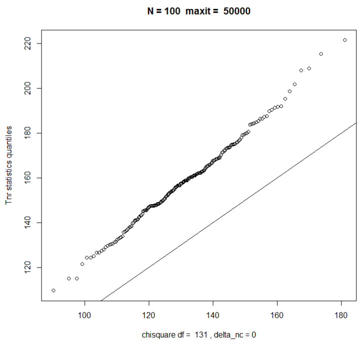

Figure 29

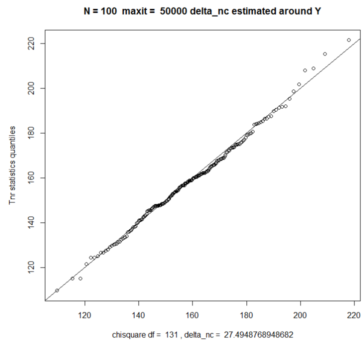

Figure 30

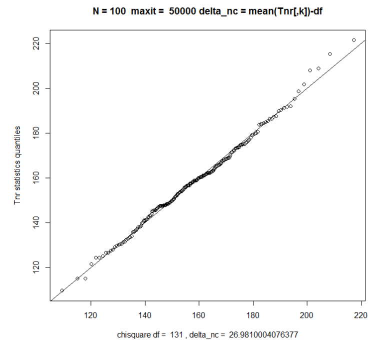

Figure 31

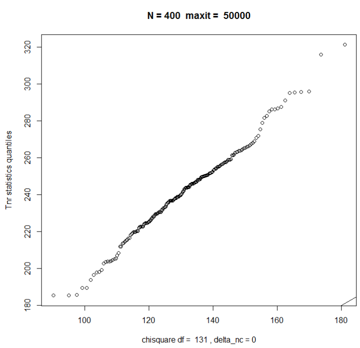

Figure 32

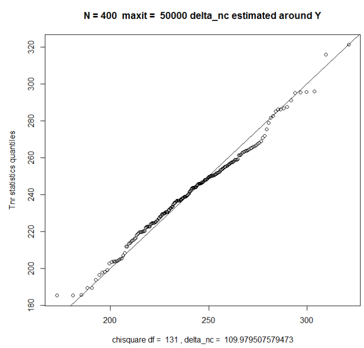

Figure 33

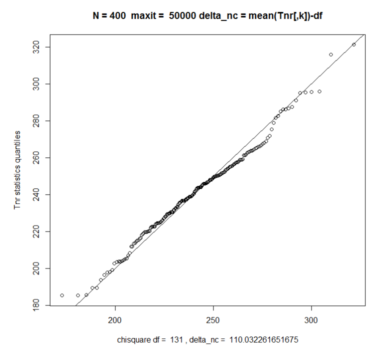

Figure 34

- Non central case: $\Delta = 4, \theta = 20, \sigma = 5,\ max\ iteration = 5 * 10^4$. In noncentral case we don't need to estimate δ_r^{opt}, because it is included in the δ_r

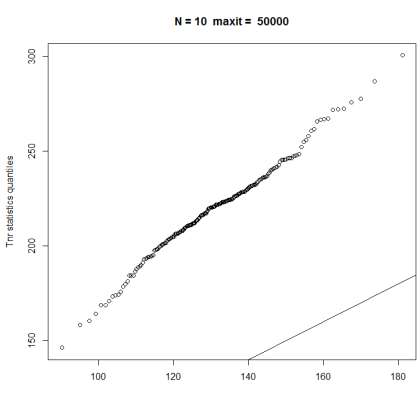

Figure 35: plot it against central chi-square

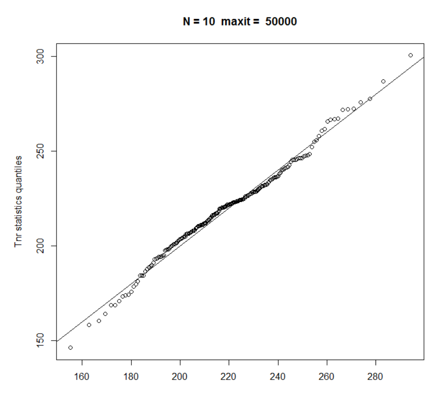

Figure 36: plot it against non-central chi-square

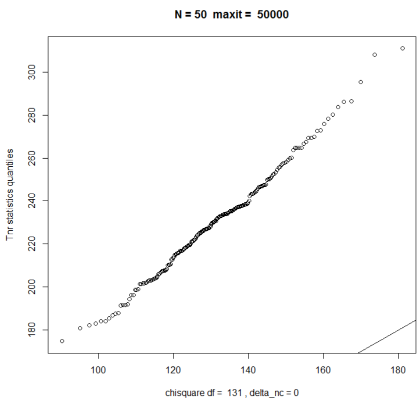

Figure 37

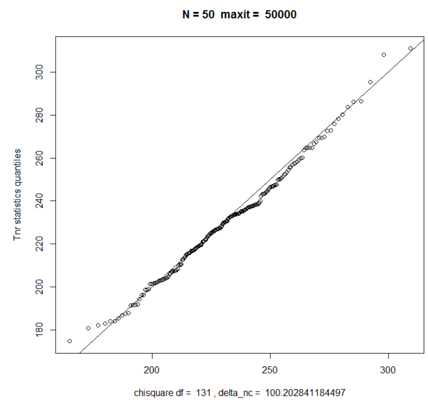

Figure 38

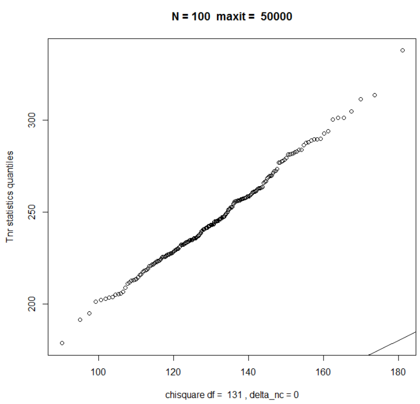

Figure 39

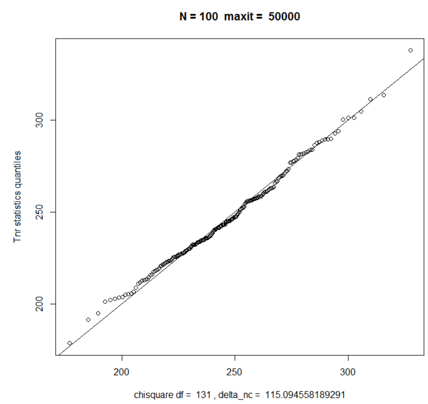

Figure 40

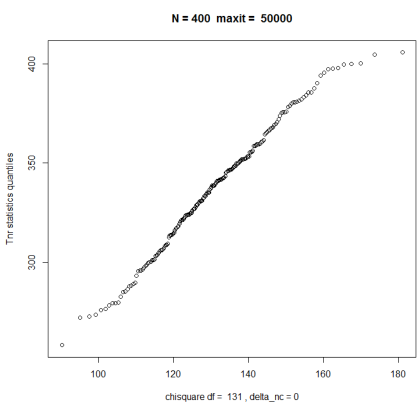

Figure 41

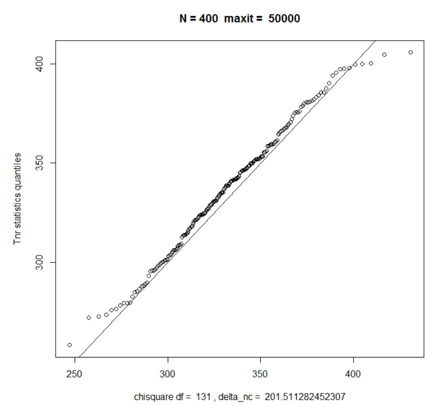

Figure 42

2 Chi-square distribution of $T_N(r,\Omega') - T_N(r,\Omega)$

2.1 Experiment Setting

- N: sample size.
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1\times n_2}$: true value. Generate $U \in \mathbb{R}^{n_1\times r}$, $V \in \mathbb{R}^{n_2\times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-20,20]$. Orthonormalize U and V to get $\widetilde{U}$ and $\widetilde{V}$, respectively. Generate D, which is $r\times r$ diagonal matrix. $Y^* = \widetilde{U}D\widetilde{V}^T$.
- Δ_{ij}: population drift.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0,\sigma^2)$, we don't need a large N to guarantee the convergence.
- Ω : set of observed positions.
- Ω': set of larger observed positions. $\Omega \subset \Omega'$.
- M_{ij}: observed values. $M = P_{\Omega'}(Y^* + N^{-1/2}\Delta + \varepsilon)$.
- $T_N(r,\Omega)$: test statistics. $T_N(r,\Omega) := N \min\limits_{Y\in\mathcal{M}_r} \sum\limits_{(i,j)\in\Omega} w_{ij}(Y_{ij} - M_{ij})^2$.
- w: weight. $w_{ij} = 1/\sigma^2$.
- δ_r: theoretical noncentrality parameter. $\delta_r = \min\limits_{H\in P_\Omega(\mathcal{T}_{\mathcal{M}_r})} \sum\limits_{(i,j)\in\Omega} \sigma_{ij}^{-2}(\Delta_{ij} - H_{ij})^2$.
- In this experiment, $n_1 = 40$, $n_2 = 50, rank(Y^*) = 11, |\Omega'| = 1001$ and the generic bound is 13.

For one N, $Y*$, Δ, Ω and Ω', multiple $\varepsilon's$ are generated (say, N_{rep} times). Then, I solved the least square problem and get $T_N(r,\Omega') - T_N(r,\Omega)$, N_{rep} times. Qqlot all N_{rep} differences with corresponding chi-squares quantiles. In this experiment $N_{rep} = 200$.

2.2 Results

All the results are based on converged central cases.

- Case 1: $|\Omega| = 1000$, $\Delta = 0$, $\sigma = 5$, $maxiteration = 10^5$, $df_{r,\Omega'} - df_{r,\Omega} = 1001 - 1000 = 1$, $\delta_{r,\Omega'} - \delta_{r,\Omega} = 0$

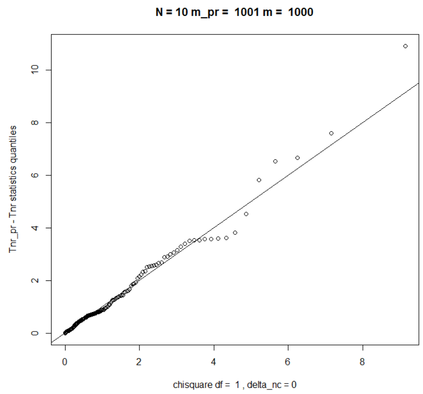

Figure 43: It fits well with the Chi-square distribution(df=1)

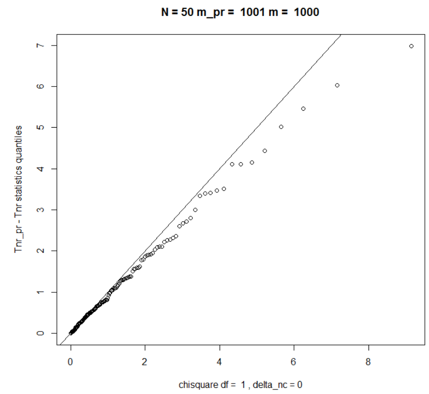

Figure 44

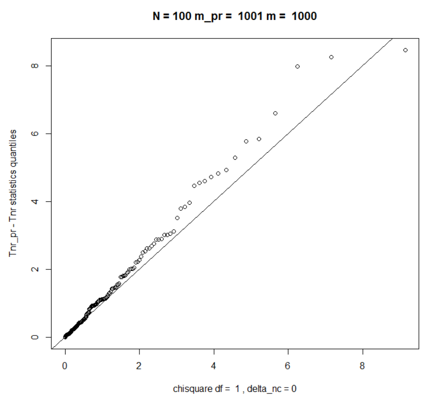

Figure 45

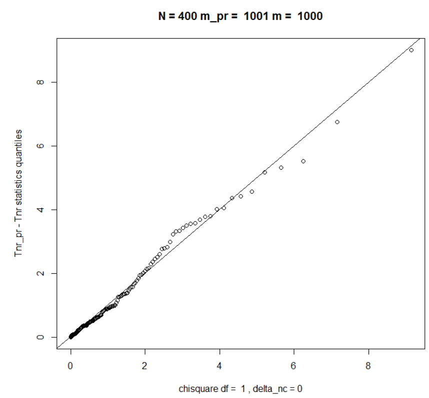

Figure 46

- Case 2: $|\Omega| = 996$, $\Delta = 0$, $\sigma = 5$, $maxiteration = 10^5$, $df_{r,\Omega'} - df_{r,\Omega} = 1001 - 996 = 5$, $\delta_{r,\Omega'} - \delta_{r,\Omega} = 0$

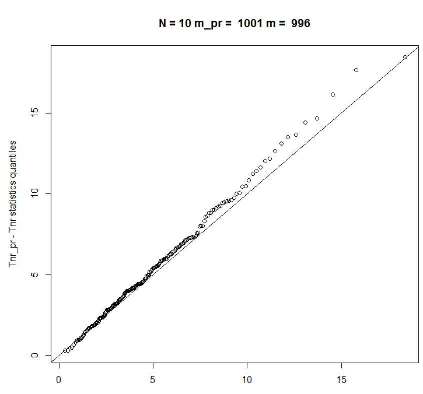

Figure 47: It fits well with the Chi-square distribution(df=5)

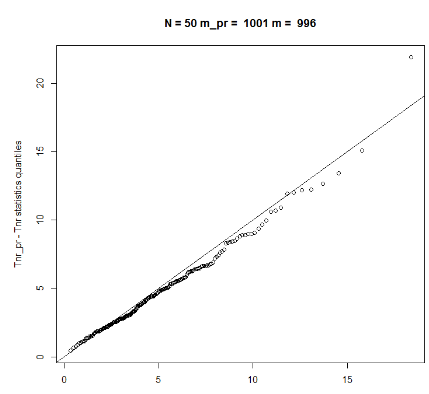

Figure 48

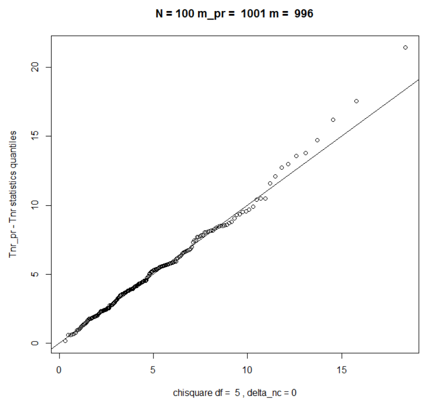

Figure 49

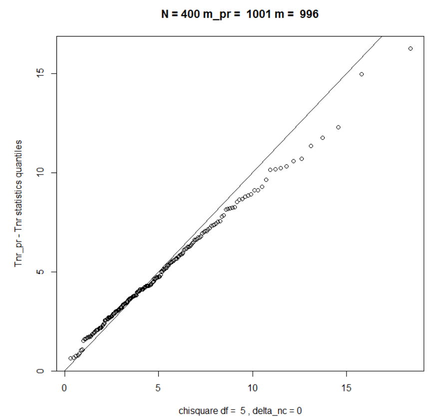

Figure 50

3 Exploration of differences between test statistics with different ranks

For this part, I am going to show what will happen if rank is misspecified (smaller or larger than the true rank of Y^*)

3.1 Experiment Setting

- N: sample size
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1 \times n_2}$: true value. $Y^* = UV^T$, where $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-20, 20]$.
- Δ_{ij}: population drift equals to 0 in this experiment.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0, \sigma^2)$, we don't need a large N to guarantee the convergence. $\sigma = 5$ in this experiment.
- Ω : set of observed positions.
- M_{ij}: observed values. $M = P_\Omega(Y^* + N^{-1/2}\Delta + \varepsilon)$.
- $T_N(r)$: test statistics. $T_N(r) := N \min\limits_{Y \in \mathcal{M}_r} \sum\limits_{(i,j)\in\Omega} w_{ij}(Y_{ij} - M_{ij})^2$
- df_r: degree of freedom. $df_r = m - r(n_1 + n_2 - r)$
- w: weight. $w_{ij} = 1/\sigma^2$
- δ_r: theoretical noncentrality parameter. $\delta_r = \min\limits_{H \in P_\Omega(\mathcal{T}_{\mathcal{M}_r})} \sum\limits_{(i,j)\in\Omega} \sigma_{ij}^{-2}(\Delta_{ij} - H_{ij})^2$
- In this experiment, $n_1 = 40,\ n_2 = 50, rank(Y^*) = 8, |\Omega| = 1000$ and $\mathfrak{R}(n_1, n_2, m) = 12$.

In this experiment, algorithm is run with r equals from 6 to 10. $\forall r_1 < r_2, T_N(r_1) > T_N(r_2)$, because we can think large r gives us a more complicate model and small r gives us a simple model. Therefore, for $r < 8, T_N(r) - T_N(8) iscomputed$ and for $r > 8, T_N(8) - T_N(r) iscomputed.$

3.2 Results

- When $r < 8$, $T_N(r) - T_N(8)$ fits Chi-square distribution with degrees of freedom $df_r - df_8$ and noncentrality parameter δ_r.
- When $r > 8$, $T_N(8) - T_N(r)$ does not fit Chi-square distribution with degrees of freedom $df_8 - df_r$ and noncentrality parameter δ_r.

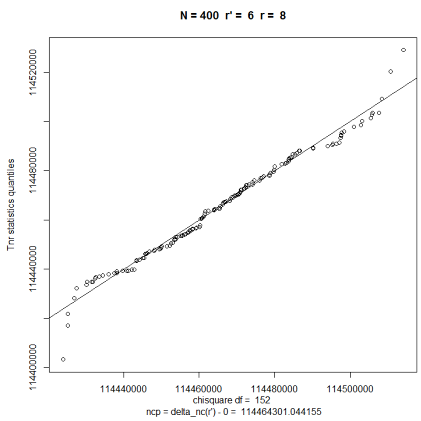

Figure 51: The difference of test statistics fits Chi-square well when r is small. We can notice the noncentrality parameter is quite huge.

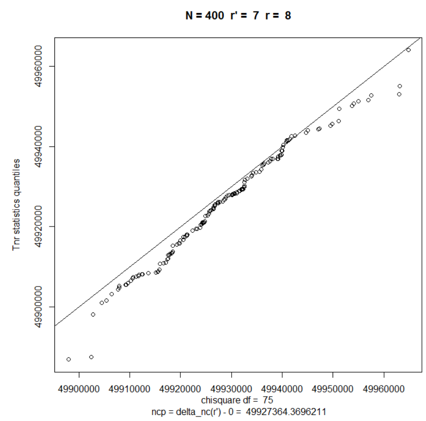

Figure 52

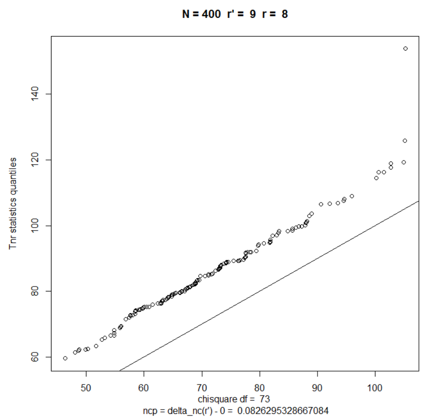

Figure 53: The difference of test statistics does not fit Chi-square well when r is large. It seems the noncentrality parameter is not correct.

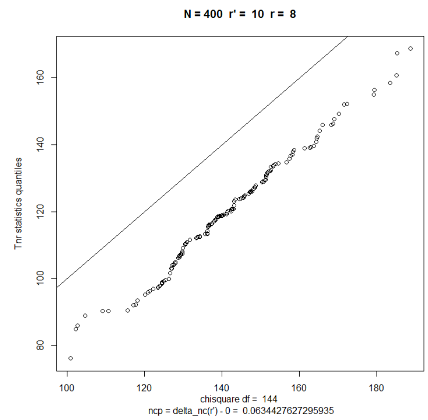

Figure 54

4 Convergence of SoftImpute Method

This part, I will show how the algorithm converge to the true Y^* and M (in all positions, observed and unobserved), as r increases from less than the true rank to larger than the true rank).

4.1 Experiment Setting

- N: sample size.
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1 \times n_2}$: true value. Generate $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-20, 20]$. Orthonormalize U and V to get $\widetilde{U}$ and $\widetilde{V}$, respectively. Generate D, which is $r \times r$ diagonal matrix. $Y^* = \widetilde{U} D \widetilde{V}^T$.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0, \sigma^2)$, we don't need a large N to guarantee the convergence.
- Ω : set of observed positions.
- M_{ij}: observed values. $M = Y^* + \varepsilon$.
- $\hat{Y}$: solution of $\min\limits_{Y \in \mathcal{M}_r} \sum\limits_{(i,j) \in \Omega} (Y_{ij} - M_{ij})^2$ by SoftImpute Method.
- DiffY: Difference between $\hat{Y}$ and Y^*, $\frac{\sum_{(i,j)} (\hat{Y}_{ij} - Y^*_{ij})^2}{n_1 n_2 \sigma^2}$.
- DiffM: Difference between $\hat{Y}$ and M, $\frac{\sum_{(i,j)} (\hat{Y}_{ij} - M_{ij})^2}{n_1 n_2 \sigma^2}$.

I generate one Y^* with Rank 9($\mathfrak{R}(n_1, n_2, m) = 12$), then generate N_{rep}'s ε, so I get N_{rep} M's(corresponding to one Y^*). For each M, SoftImpute is applied, with r specified from 7 to 11. $N_{rep} = 100$

4.2 Results

- Under this experiment setting, algorithm converge at rank 8,9; doesn't converge at rank 10, 11 very few times, and don't converge at rank 7 all the time.
- When r is specified as the true rank, we get the least DiffY and DiffM. When r is mis-specified, DiffY and DiffM increases. DiffY and DiffM increases hugely when r is mis-specified to be less than the true rank.

	rank = 7	rank = 8	rank = 9	rank = 10	rank = 11
Min.	74694.28	3029.044	0.1578254	57.38970	87.65924
1st Qu.	75451.40	3044.827	0.2444765	58.87480	92.76772
Median	75753.57	3050.506	0.3092179	59.78151	96.15876
Mean	75787.34	3050.042	0.3159677	59.77329	96.16106
3rd Qu.	76131.67	3053.852	0.3874277	60.53845	99.54820
Max.	77588.92	3076.948	0.5809751	63.89462	105.45933

Figure 55: Summary of DiffY at different ranks. N=10

	rank = 7	rank = 8	rank = 9	rank = 10	rank = 11
Min.	75574.26	3046.015	0.02790767	58.86475	93.61858
1st Qu.	75760.79	3048.293	0.08249270	59.21434	94.55833
Median	75807.80	3048.886	0.09891904	59.30534	94.96089
Mean	75805.55	3049.056	0.09924143	59.30929	94.94210
3rd Qu.	75859.30	3049.924	0.12048925	59.43047	95.31139
Max.	76126.99	3052.479	0.16180806	59.71012	96.63756

Figure 56: Summary of DiffY at different ranks. N=400

	rank = 7	rank = 8	rank = 9	rank = 10	rank = 11
Min.	74695.53	3029.092	0.1891477	57.24998	87.74554
1st Qu.	75453.60	3044.666	0.2734890	58.90999	92.76938
Median	75751.11	3050.573	0.3371112	59.82306	96.10776
Mean	75787.81	3050.242	0.3423473	59.77354	96.15791
3rd Qu.	76132.76	3054.487	0.4130562	60.53634	99.62054
Max.	77587.66	3077.533	0.6073559	63.99402	105.54641

Figure 57: Summary of DiffM at different ranks. N=10

	rank = 7	rank = 8	rank = 9	rank = 10	rank = 11
Min.	75573.83	3045.947	0.02857006	58.86324	93.64121
1st Qu.	75761.90	3048.229	0.08297967	59.22046	94.55679
Median	75808.30	3048.989	0.09889706	59.30971	94.96230
Mean	75805.74	3049.065	0.10013196	59.31355	94.94737
3rd Qu.	75859.37	3049.944	0.12142724	59.42805	95.30558
Max.	76127.51	3052.385	0.16313486	59.72817	96.65499

Figure 58: Summary of DiffM at different ranks. N=400

5 Probability to satisfy the necessary condition(Proposition 1.3)

5.1 Experiment Setting

- n_1: number of rows.
- n_2: number of columns.
- m: number of observations. In this experiment m increase from 100 to $n_1 * n_2$
- $\mathfrak{R}(n_1, n_2, m)$: generic bound. $\mathfrak{R}(n_1, n_2, m) = (n_1 + n_2)/2 - \sqrt{(n_1 + n_2)^2/4 - m}$

In this experiment, I will show, first, for a given m, how the probability to satisfy the necessary condition drops when r is increasing; second, for a given m, the probabilities when r is chosen to be smaller than the $\mathfrak{R}(n_1, n_2, m)$.

5.2 Results

- For each m(not too small), as r increases, the probability will maintain at close to 100% for some ranks, and then drop to near 0% quickly.
- When the matrix is closed to square matrix, for large m, if r is chosen to be small enough (say, 1 or 2 less than $\mathfrak{R}(n_1, n_2, m)$), the probability is near 100%

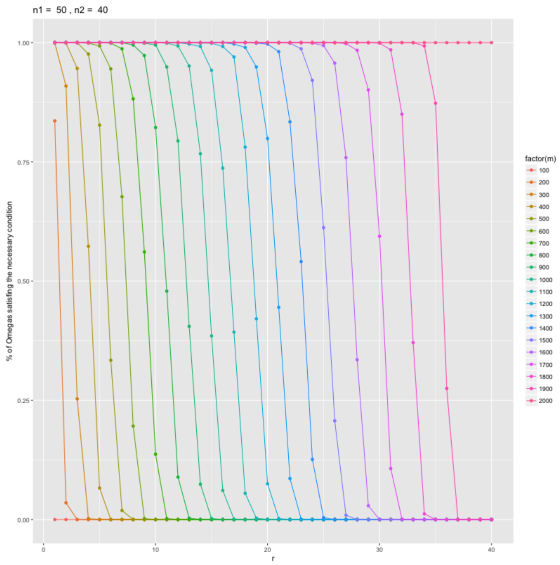

Figure 59: Probability of a given m for any r

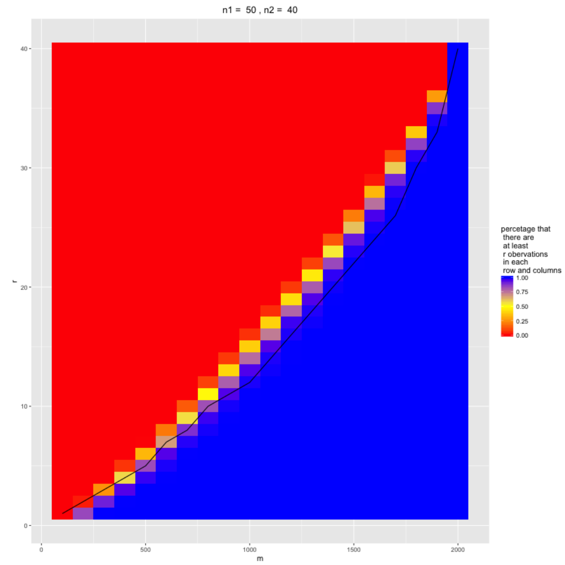

Figure 60: The line in the middle is the $\mathfrak{R}(n_1, n_2, m)$. For m large enough, the probability is very close to 100%, for any r less than $\mathfrak{R}(n_1, n_2, m)$

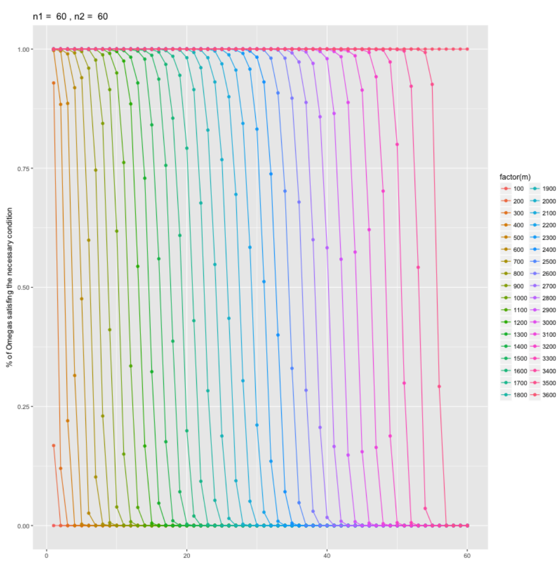

Figure 61

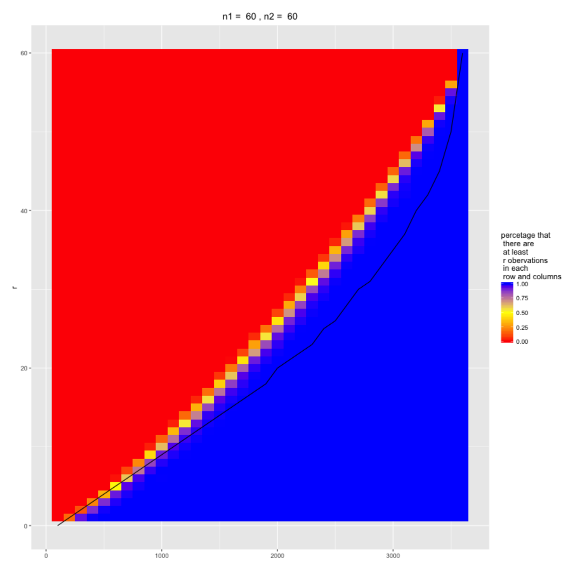

Figure 62

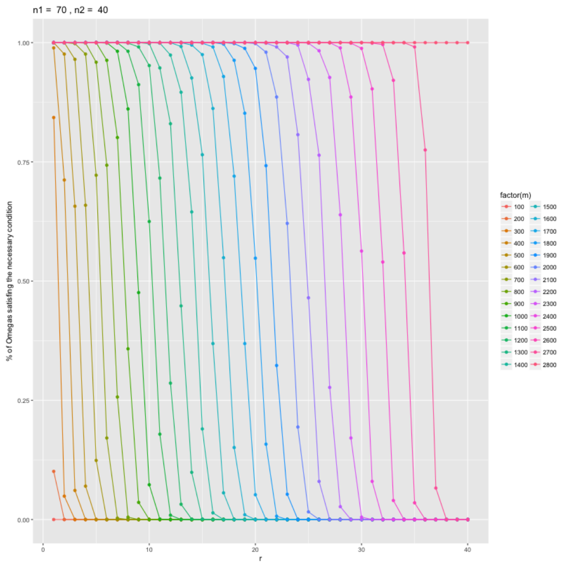

Figure 63

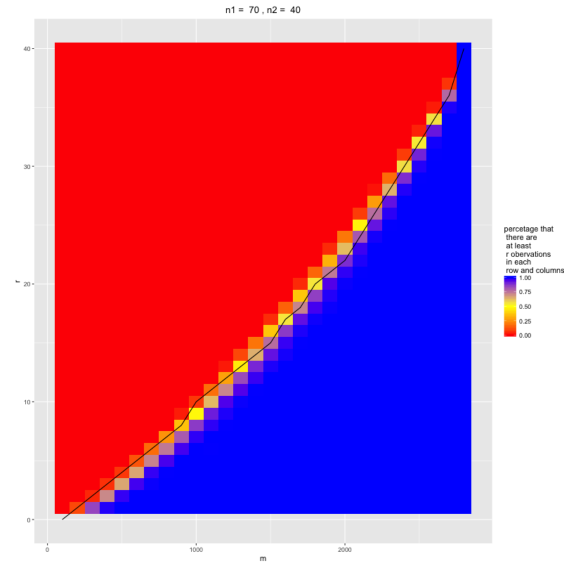

Figure 64: For a not close-to-square matrix, even large m can't not guarantee high probability for any r less than $\Re(n_1, n_2, m)$.

6 Exploration of matrix not having locally unique solution

6.1 Experiment Setting

- N: sample size.
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1 \times n_2}$: true value. Generate $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-20, 20]$. Orthonormalize U and V to get $\widetilde{U}$ and $\widetilde{V}$, respectively. Generate D, $D_{ii} \sim Uniform[5000, 10000]$, which is $r \times r$ diagonal matrix. $Y^* = \widetilde{U} D \widetilde{V}^T$.
- Δ_{ij}: population drift.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0, \sigma^2)$, we don't need a large N to guarantee the convergence.
- Ω : set of observed positions.
- M_{ij}: observed values. $M = P_\Omega(Y^* + N^{-1/2}\Delta + \varepsilon)$.
- $T_N(r)$: test statistics. $T_N(r) := N \min_{Y \in \mathcal{M}_r} \sum_{(i,j)\in\Omega} w_{ij}(Y_{ij} - M_{ij})^2$.
- w: weight. $w_{ij} = 1/\sigma^2$.
- δ_r: theoretical noncentrality parameter. $\delta_r = \min_{H \in P_\Omega(\mathcal{T}_{\mathcal{M}_r})} \sum_{(i,j)\in\Omega} \sigma_{ij}^{-2}(\Delta_{ij} - H_{ij})^2 \approx N \min_{Y \in \mathcal{M}_r} \sum_{(i,j)\in\Omega} w_{ij}(Y^*_{ij} + N^{-1/2}\Delta_{ij} - Y_{ij})^2$.

type two algorithms are implements, `type="svd"` or the default `type="als"`. The "svd" algorithm repeatedly computes the svd of the completed matrix, and soft thresholds its singular values. Each new soft-thresholded svd is used to re-impute the missing entries. For large matrices of class `"Incomplete"`, the svd is achieved by an efficient form of alternating orthogonal ridge regression. The "als" algorithm uses this same alternating ridge regression, but updates the imputation at each step, leading to quite substantial speedups in some cases. The "als" approach does not currently have the same theoretical convergence guarantees as the "svd" approach.

Figure 65: This is the description in the document of the package.

- method: "svd" and "als". In all experiment(not only this chapter), if it is not specified, the method I used is "svd" because it has theoretical convergence guarantees. However, when deal with the reducible matrix, "svd" can't give us results(algorithm stops at the first iteration). Detail of this two method is showed in the following pictures.

6.2 Results

6.2.1 Not having r observations in each row and column

- For the matrices don't satisfy this necessary condition, we can have a way to quantify how "severe" they violate this condition. For example, both A and B violate the condition only in one column. But A has $r-1$ observations on the column and B only has $r-5$ observations on the column. We can say, B violates the condition more severe than A. Generally speaking, for a matrix A, we can define

$$SE(A) = \sum_{\substack{rows\, or\, columns \\ not\, having\, r\, observations}} (r - \#\, of\, observations)$$

  to quantify the severity.

- From the results, it seems the noncentrality parameter of $T_N(r)$ becomes larger when SE(A) becomes larger.

- For this part, "svd" can still give us reasonable results, therefore I still use "svd" in this part.

- In the three experiments of following three plots, Y^*'s are the same.

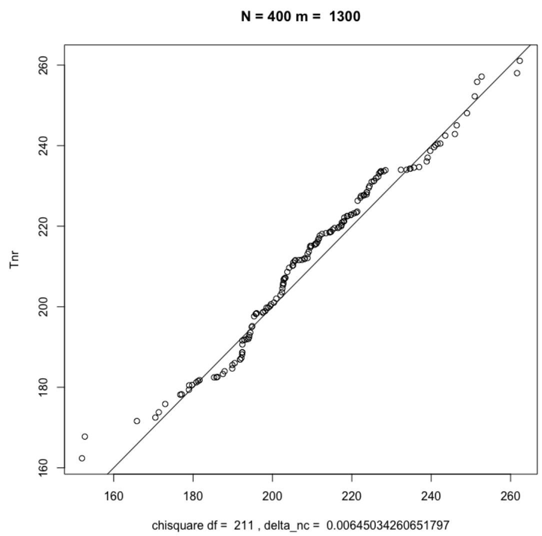

Figure 66: $n_1 = 70, n_2 = 40, m = 1300, r = 11, \mathfrak{R}(n_1, n_2, m) = 13$. One row violates the condition, $SE(A) = 2$. The estimated noncentrality parameter is close to 0 (which should be 0 theoretically). We can see it is still close to the Chi-square distribution.

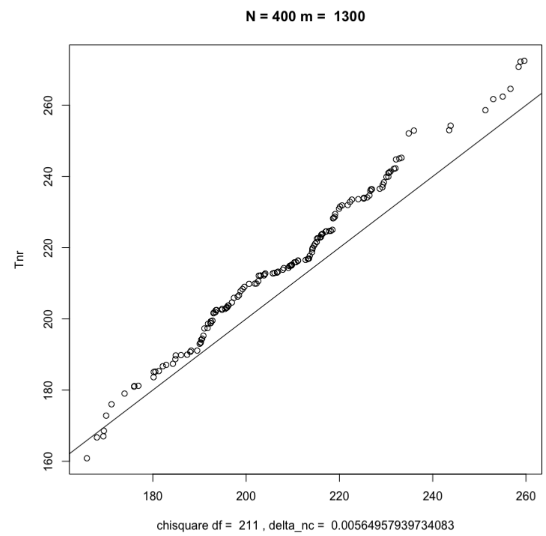

Figure 67: $n_1 = 70, n_2 = 40, m = 1300, r = 11$. Three rows violate the condition, $SE(A) = 4$. The estimated noncentrality parameter is close to 0. We can see most of the quantiles of $T_N(r)$ are larger than the quantiles of the Chi-square distribution.

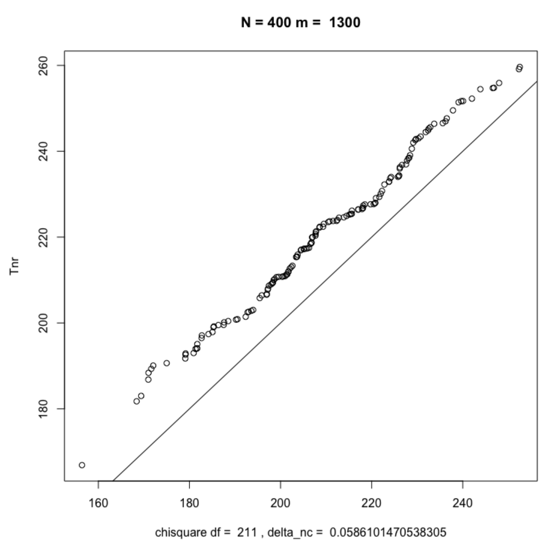

Figure 68: $n_1 = 70, n_2 = 40, m = 1300, r = 11$. Five rows violate the condition, $SE(A) = 8$. The estimated noncentrality parameter is close to 0. We can see the discrepancy becomes clearer as SE(A) increases.

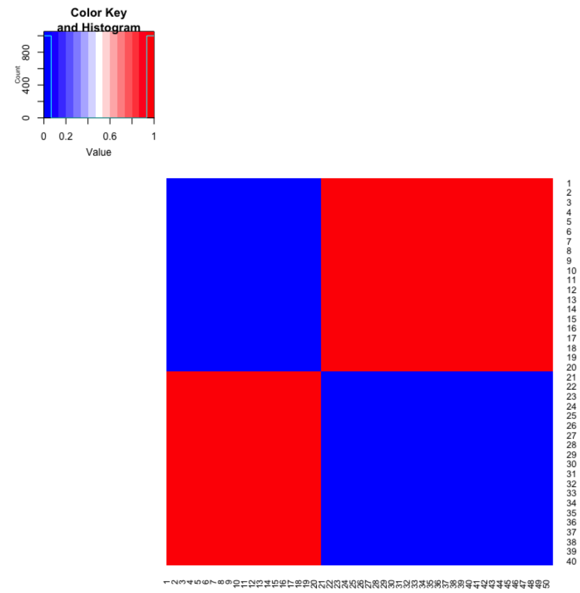

Figure 69: Heapmap of average absolute error of recovering Y^* with $P_\Omega(Y^*)$, $n_1 = 40, n_2 = 50, m = 1000, LT \in \mathbb{R}^{20\times 20}, RB \in \mathbb{R}^{30\times 30}, r = 10$. This matrix satisfies necessary condition but does not satisfy the sufficient condition. We can see the error of observed part is close to zero. The error larger than 1 is truncated to be 1, the error of unobserved part is acctually huge.

6.2.2 Reducible case

- In this part, the Ω has the following form:

$$P_\Omega(M) = \begin{bmatrix} LT & 0 \\ 0 & RB \end{bmatrix}$$

- For the above case, even though it satisfies the necessary condition, it is still not locally unique. "svd" can't be used to solve this kind of problem. For this part, I use "als" to solve the optimization problem.

- If Y^* is recovered through $P_\Omega(Y^*)$, the left top and right bottom can be recovered correctly, however, for the unobserved part, the error is huge(Figure69). This result is reasonable. The true rank of Y^* is r and the left top and right bottom are observed, which should guarantee the accuracy of the recovery of them. However, the unobserved part is unindentifiable.

- If Y^* is recovered by observation with noise $P_\Omega(M)$, the results on the observed position are incorrect as well. (Probably, the algorithm converges to another matrix with r = 10. This could be an evidence of the unstablility even on the observed position.) To show this, we can just show the qqplots again(Figure70,71,72,73).

- To be cautious, the Figure74 shows the discrepancy is not due to the method "als".

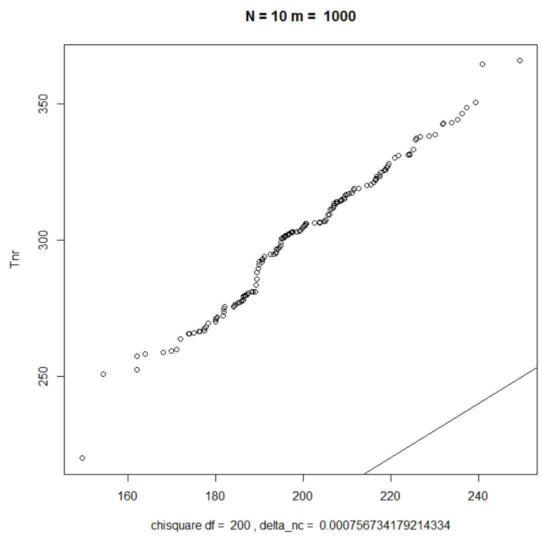

Figure 70: Y^* and Ω are the same as the Figure 69. The estimated noncentrality parameter is very small, which consistent with the above conclusion, since the estimated noncentrality parameter is just the error of the observed position through $P_\Omega(Y^*)$. Large discrepancy between $T_N(r)$ and Chi-square distribution shows that the recovery on observed position through $P_\Omega(M)$ is incorrect.

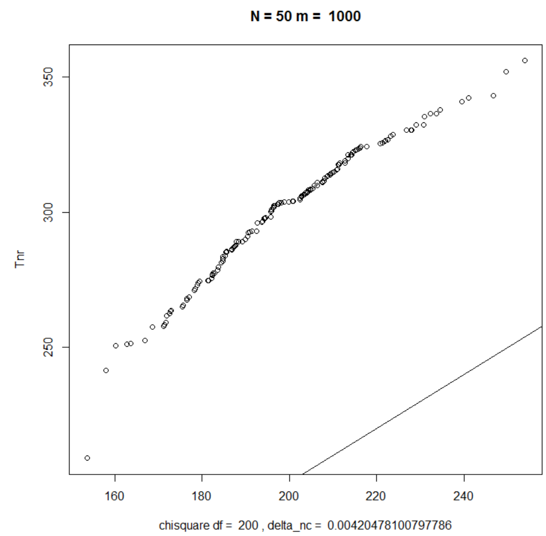

Figure 71

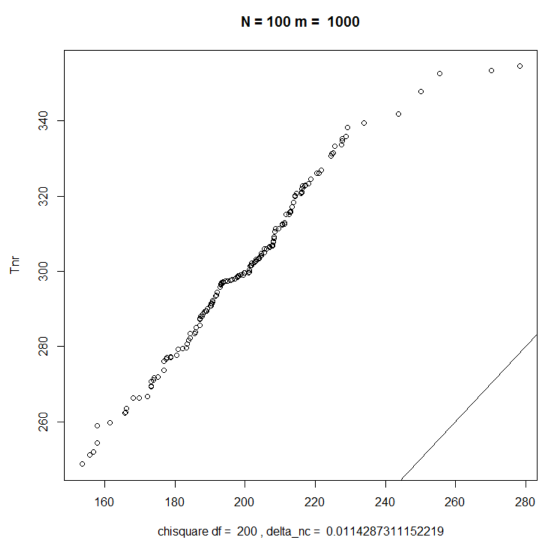

Figure 72

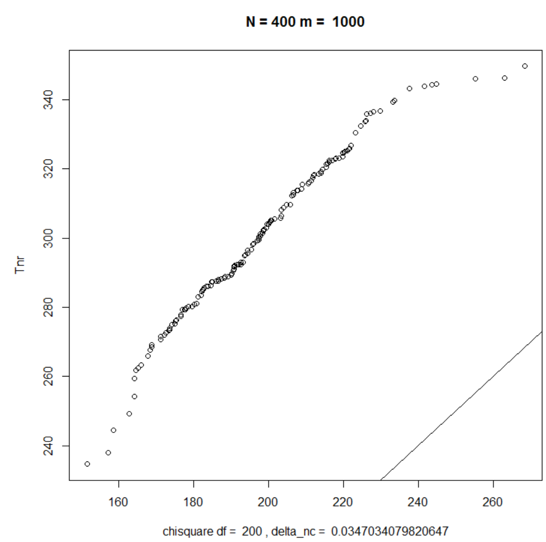

Figure 73

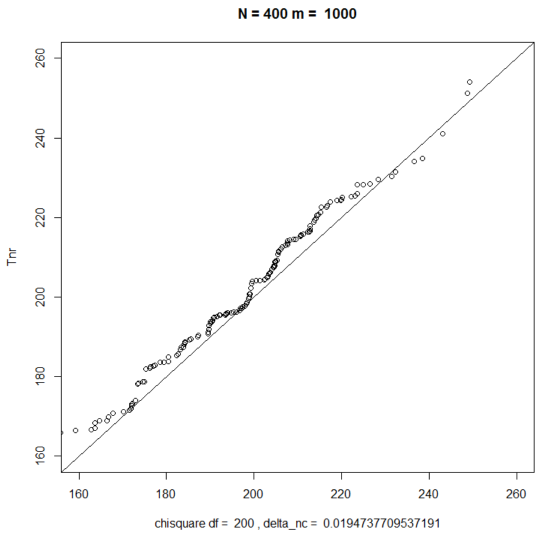

Figure 74: Y^* is the same as previous example, however the Ω is randomly sampled and satisfy the sufficient condition. We can see "als" do its job as well as "svd". This gives us evidence that although "als" does not guarantee theoretical convergence, it does not cause the discrepancy in Figure 70, 71, 72, 73

7 Rank test

7.1 Experiment Setting

- N: sample size.
- $Y^* \in \mathcal{M}_r \subset \mathbb{R}^{n_1 \times n_2}$: true value. Generate $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{n_2 \times r}$, $\forall U_{ij}, V_{ij} \sim Uniform[-20, 20]$. Orthonormalize U and V to get $\widetilde{U}$ and $\widetilde{V}$, respectively. Generate D, $D_{ii} \sim Uniform[5000, 10000]$, which is $r \times r$ diagonal matrix. $Y^* = \widetilde{U} D \widetilde{V}^T$.
- Δ_{ij}: population drift. $\Delta_{ij} = 0$ in this experiment.
- ε_{ij}: random errors. $\varepsilon_{ij} \sim N(0, \frac{\sigma^2}{N})$. Note that $\forall N$, $N^{1/2}\varepsilon_{ij} \sim N(0, \sigma^2)$, we don't need a large N to guarantee the convergence. $\sigma = 5$ in this experiment.
- Ω : set of observed positions.
- M_{ij}: observed values. $M = P_\Omega(Y^* + N^{-1/2}\Delta + \varepsilon)$.
- $T_N(r)$: test statistics. $T_N(r) := N \min\limits_{Y \in \mathcal{M}_r} \sum\limits_{(i,j) \in \Omega} w_{ij}(Y_{ij} - M_{ij})^2$.
- w: weight. $w_{ij} = 1/\sigma^2$.

- δ_r: theoretical noncentrality parameter. $\delta_r = \min_{H \in P_\Omega(\mathcal{T}_{\mathcal{M}_r})} \sum_{(i,j)\in\Omega} \sigma_{ij}^{-2}(\Delta_{ij} - H_{ij})^2 \approx N \min_{Y \in \mathcal{M}_r} \sum_{(i,j)\in\Omega} w_{ij}(Y^*_{ij} + N^{-1/2}\Delta_{ij} - Y_{ij})^2$.

- method: ”svd”.

Normally, rank test test starts from $r = 1$ to $r = \mathfrak{R}(n_1, n_2, m)$. From [2], we know, if the unobserved entries are repalced with 0, the underlying structure of matrix can still be seen. Therefore, by looking into svd of M we might be able to narrow the range of r.

7.2 Result

- Doing the hypothesis test with H_0: rank is $r(r = 1 \cdots \mathfrak{R}(n_1, n_2, m))$, choosing the smallest r where H_0 is not rejected, we can detect the true rank with large probability. I tried this procedure 200 times with the case, which $n_1 = 40, n_2 = 50, m = 1000, rank = 9, N = 400$, and the true rank can be detected with 92.5%.

- H_0 is not rejected only when r is closed to true rank.

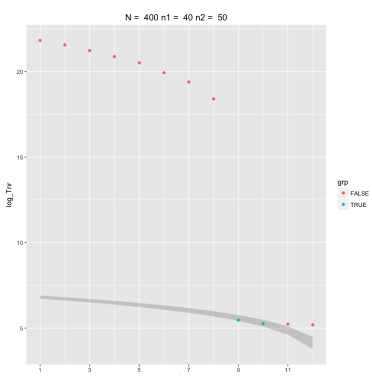

Figure 75: The grey region is the confidence interval with $\alpha = 0.05$. The red points are not in the confidence interval while the blue points are in the confidence interval. To better show the data, I plot it in log scale. In this case, the true rank is 9.

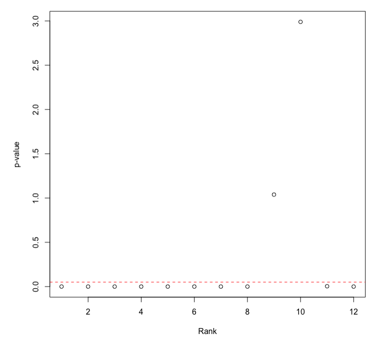

Figure 76: This is p-value for the test on each rank. The red dash line is p-value = 0.025, since these tests are two-sided tests. This plot is equivalent to Figure 75.

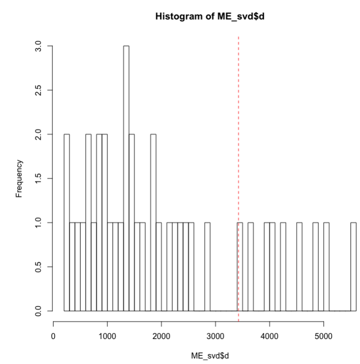

Figure 77: This is histogram of the singular values of M. The number of singular value on the right of red dashed line is equal to the true rank. We can start the sequential rank test from the r where the gap is large. In this case, it can be r = 1, 3, 4, 7

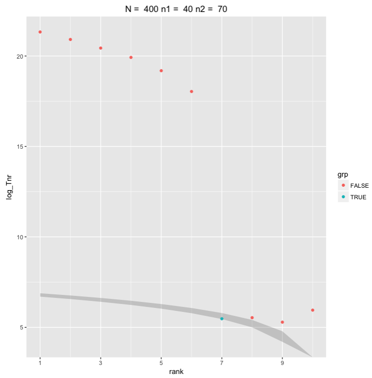

Figure 78: True rank is 7.

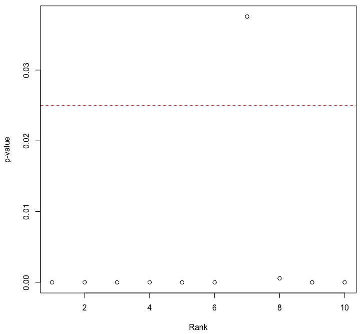

Figure 79

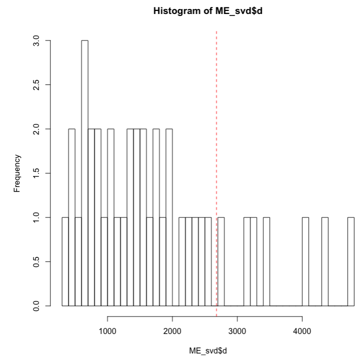

Figure 80

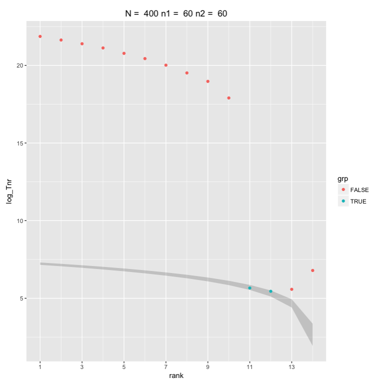

Figure 81: The true rank is 11.

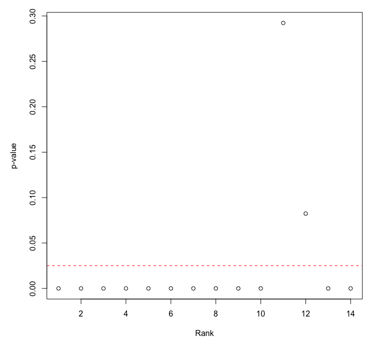

Figure 82

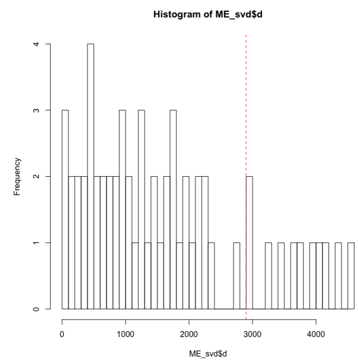

Figure 83

8 Study of the 6 × 6 old sample

8.1 Result

- The two solutions provided in [3] are indeed rank 3 matrices.
- Soft-thresholded SVD can't recover any of these two solutions. The algorithm converges to another solution. This method is not an "exact" method, therefore this result make sense.
- I also tried nuclear norm minimization. It also can't recover any of these two solution. The algorithm converge to a rank 4 matrix. An interesting finding is that the nuclear norm of this rank 4 matrix is smaller than those two rank 3 matrices.

The two solutions provided in [3]:

$$\begin{pmatrix} 0.64 & 0.56 & 0.16 & 0.48 & 0.24 & 0.64 \\ 0.56 & 0.85 & 0.20 & 0.66 & 0.51 & 0.86 \\ 0.16 & 0.20 & 0.06 & 0.18 & 0.07 & 0.23 \\ 0.48 & 0.66 & 0.18 & 0.56 & 0.3 & 0.72 \\ 0.24 & 0.51 & 0.07 & 0.30 & 0.50 & 0.41 \\ 0.64 & 0.86 & 0.23 & 0.72 & 0.41 & 0.93 \end{pmatrix}$$

$$\begin{pmatrix} 0.425616 & 0.56 & 0.16 & 0.48 & 0.24 & 0.64 \\ 0.56 & 0.902308 & 0.20 & 0.66 & 0.51 & 0.86 \\ 0.16 & 0.20 & 0.063469 & 0.18 & 0.07 & 0.23 \\ 0.48 & 0.66 & 0.18 & 0.546923 & 0.3 & 0.72 \\ 0.24 & 0.51 & 0.07 & 0.30 & 0.386667 & 0.41 \\ 0.64 & 0.86 & 0.23 & 0.72 & 0.41 & 0.998 \end{pmatrix}$$

The solution of Soft-thresholded SVD:

$$\begin{pmatrix} 0.4103942 & 0.5595528 & 0.15171237 & 0.4831419 & 0.24001763 & 0.6399771 \\ 0.5595528 & 0.7581729 & 0.20728691 & 0.6580932 & 0.50997645 & 0.8600077 \\ 0.1517124 & 0.2072869 & 0.05601268 & 0.1785344 & 0.07018296 & 0.2299812 \\ 0.4831419 & 0.6580932 & 0.17853440 & 0.5681645 & 0.29995506 & 0.7200167 \\ 0.2400176 & 0.5099764 & 0.07018296 & 0.2999551 & -6.99395554 & 0.4100003 \\ 0.6399771 & 0.8600077 & 0.22998119 & 0.7200167 & 0.41000028 & -0.922015 \end{pmatrix}$$

The solution of nuclear norm minimization:

$$\begin{pmatrix} 0.4369 & 0.56 & 0.16 & 0.48 & 0.24 & 0.64 \\ 0.56 & 0.7625 & 0.20 & 0.66 & 0.51 & 0.86 \\ 0.16 & 0.20 & 0.0520 & 0.18 & 0.07 & 0.23 \\ 0.48 & 0.66 & 0.18 & 0.5302 & 0.3 & 0.72 \\ 0.24 & 0.51 & 0.07 & 0.30 & 0.1926 & 0.41 \\ 0.64 & 0.86 & 0.23 & 0.72 & 0.41 & 0.9555 \end{pmatrix}$$

References

[1] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[2] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[3] E. B. Wilson and J. Worcester, "The resolution of six tests into three general factors," *Proceedings of the National Academy of Sciences*, vol. 25, no. 2, pp. 73–77, 1939.