Character Analysis of *Dream of the Red Chamber*: A Dimension Reduction Perspective

Chiyu Ma
Department of Applied Mathematics
Hong Kong Polytechnic University
chiyu.ma@connect.polyu.hk

Yuqia Wu
Department of Applied Mathematics
Hong Kong Polytechnic University
yuqia.wu@connect.polyu.hk

Abstract

Dream of the Red Chamber is an invaluable Chinese novel appealing researchers from different fields. Given character-event data, we analyze relation of characters by dimension reduction and study the plots by sparse principal component analysis. We visualize the affinity among the key characters by non-metric MDS and give some tables to illustrate the main characters and plots of the novel.

1 Introduction

As one of the most popular Chinese novel, *Dream of the Red Chamber* attracts not only literature scholars, but also statisticians. At the beginning, statisticians regarded words and sentences as original data and some interesting results have been got. Compared with language analysis, statistical analysis is not widely applied in character relations. Given the character-event data of *Dream of the Red Chamber*, we can visualize character relation in a more statistical way. And it's also a good data set to test dimension reduction techniques. In this report, we establish a dissimilarity matrix and visualize the relation among key characters by non-Metric MDS. And we also use principal component analysis to analyze the main roles of the two parts (1-80 chapters and 81-120 chapters) in the book and show the difference between these two parts.

2 Data

The data set we use is from `https://github.com/yuany-pku/dream-of-the-red-chamber`, which is a 376×475binary matrix. Rows of this matrix is characters and columns are events. If character$i$ appears in event$j$, the element ij is 1. As [1] pointed out, it contains 2 pairs of duplicate rows. After remove the duplicate ones, we have a 374×475 matrix. Plus, there are 55 characters involved in nothing. Thus we can deal with the smaller matrix which is 319×475[1]. It is commonly believed the last 40 chapters' author is different from the first 80 chapters'. Thus we divide it to two sub-matrices, which are 319×350 and 319×125.

3 Visualization of Characters' Relation by non-Metric MDS

The most common character relation diagram is based on blood relation. And generally it is two tree, RongGuoFu荣国府 and NingGuoFu宁国府. This kind of diagram helps readers to understand

[1] Besides these two error, we suspect the original data has more mistakes. For example, the last event *Meeting near the river*江头遇故人 is a meeting between JiaZheng 贾政 and JiaBaoyu贾宝玉. However neither of these two row is one in the last column.But we don't know how the raw data was collated.Perhaps the standards for collating data are more stringent.We made no other revisions to the original data.

Preprint. Under review.

characters' genetic relationship, but it doesn't reflect the intimate degree among them. By non-metric MDS, we can construct a diagram to remedy the limitation.

3.1 non-Metric MDS

Classic or metric MDS is to build points' coordinates from their pair distances. But in many situation, we only have the degree of similarity or dissimilarity among the points. It is not reasonable to regard these dissimilarity as distances. Thus non-metric MDS is proposed.

Ono-metric MDS could be traced to [2, 3]. But it was perfected in [4, 5]. We simply describe the outline of it here.

Given a $n \times n$ dissimilarity matrix $\Delta = \{\delta_{ij}\}$, which is a symmetric matrix with zero diagonal, we can construct these n points' coordinates in p dimension $X \in \Re^{p\times n}$. $X = [x_1, x_2, \cdots, x_n]$ and $x_i \in \Re^p$.To visualize the points, p could be 2 or 3. d_{ij} denotes the Euclidean distance between x_i and x_j and $D = \{d_{ij}\} \in \Re^{n\times n}$. Additionally, we define such a *stress*,

$$S = \sqrt{\frac{\sum_{i<j}(f_{ij}(D,\Delta) - d_{ij})^2}{\sum_{i<j} d_{ij}^2}}.$$

$f_{ij}(D, \Delta)$ is called as disparity between i, j, which is a monotonic transformation of δ_{ij}. Specifically,

$$\delta_{ij} \le \delta_{kl} \to f_{ij}(D,\Delta) \le f_{kl}(D,\Delta), \quad \forall i, j, k, l.$$

$f_{ij}(D, \Delta)$ is also related to D. $f_{ij}(D, \Delta)$ is calculated by $\sum_{i=1}^{j} \hat{d}_i, \quad j = 1, 2, \cdots, n(n-1)/2$, where $\hat{d}_i$ is the sorted version of d_{ij}. The specific formula of $f_{ij}(D, \Delta)$ is too complicated. Readers can refer to [5].

Algorithm 1: non-Metric MDS

Data: dissimilarity matrixΔ
Result: Coordinates X
1 Initialize X;
2 Compute D and $f_{ij}(D, \Delta)$;
3 $X = \arg\min_X S$;
4 If the stopping criterion is not satisfied, go to step 2;

When compute $X = \arg\min_X S$, f_{ij} are fixed and d_{ij} is a function of x_i, x_j. Such a minimization could be got by gradient descent. Other implementation details are also in [5].

3.2 Dissimilarity Matrix of Key Characters

First, we extract the key characters from 319 characters. When extracting key characters, we use the 319×475 matrix. Then we will apply the result to two sub-matrices. Every character appears in an average of 8.7 events. Thus, if a character appears in more than 8 events, we will regard him or her as a key character. And table 1 is the list of key characters. There are 58 key characters at all.

The similarity matrix $S = \{s_{ij}\}$ is easier to define from the character-event matrix. S is initialized as a zero matrix. Every character i has a binary row of 475 events. For character i, j, if they appear in a same event, s_{ij} goes up by one. But this similarity matrix's diagonal contains different number. We can revise it in the following way, to get a revised similarity matrix $S_r = \{sr_{ij}\}$.

$$sr_{ij} = 58 \times \frac{s_{ij}}{\sqrt{s_{ii} \times s_{jj}}}.$$

Then the dissimilarity matrix $\Delta = \{\delta_{ij}\}$ is defined by

$$\delta_{ij} = 58 - sr_{ij}.$$

As we can see, the dissimilarity matrix Δ is a symmetric matrix whose diagonal elements are zero.

Table 1: Key Characters in *Dream of the Red Chamber*

Name	Number	Name	Number	Name	Number
贾赦JiaShe	25	薛姨妈XueYima	53	雪雁Xueyan	12
贾政JiaZheng	69	薛蟠XuePan	26	鸳鸯Yuanyang	35
贾珍JiaZhen	44	薛蝌XueKe	12	琥珀Hupo	12
贾琏JiaLian	70	薛宝钗XueBaochai	121	莺儿Yinger	11
贾宝玉JiaBaoyu	238	薛宝琴XueBaoqin	24	平儿Pinger	68
贾环JiaHuan	30	林黛玉LinDaiyu	118	小红Xiaohong	13
贾元春JiaYuanchun	10	邢夫人XingFuren	56	彩云Caiyun	11
贾迎春JiaYingchun	23	尤氏YouShi	37	茗烟Mingyan	14
贾探春JiaTanchun	55	李纨LiWan	57	李贵LiGui	10
贾惜春JiaXichun	33	秦氏QinShi	12	芳官Fangguan	12
贾蓉JiaRong	34	香菱Xiangling	21	贾雨村JiaYucun	10
贾兰JiaLan	20	妙玉Miaoyu	11	周瑞家的ZhouRuiJiaDe	13
贾蔷JiaQiang	11	赵姨娘ZhaoYiniang	23	秦钟QinZhong	11
贾芸JiaYun	18	刘姥姥LiuLaolao	17	赖大LaiDa	11
贾巧姐JiaQiaojie	13	袭人Xiren	101	林之孝LinZhixiao	10
史太君ShiTaijun	109	晴雯Qingwen	30	林之孝家的LinZhixiaoJiaDe	11
史湘云ShiXiangyun	46	麝月Sheyue	32	邢岫烟XingXiuyan	15
王夫人WangFuren	106	秋纹Qiuwen	18	李婶娘LiShenniang	9
王仁WangRen	9	紫鹃Zijuan	33	尤二姐YouErjie	13
王熙凤WangXifeng	134				

The original character-event matrix has 475 events. The first 80 chapters have 350 events and the last 40 chapters have 125 events. We can have three different dissimilarity matrices Δ_w, Δ_c and Δ_g(Whole, Cao and Gao[2]).And we will get three coordinate matrices X_w, X_c and X_g.

3.3 Diagram of Key Characters' Relation

We use Matlab's function `mdscale(D,p)` to get the coordinate matrices X_w, X_c and X_g. To visualize the diagram clearly, we set $p = 2$. Figure 1, Figure2 and Figure3 show our results.

These diagrams give a reasonable and interesting visualization of characters' relation. It can also tell us something that we've missed in our reading. We just list a few here.

- The relation between LinDaiyu林黛玉 and XueBaochai薛宝钗 is closer than the relations between JiaBaoyu贾宝玉 and LinDaiyu林黛玉. Due to the history background, it's reasonable.
- In the first 80 chapters, WangXifeng王熙凤, WangFuren王夫人 and ShiTaijun史太君, who are the main leaders of the family, are close. However, in the last 40 chapters, their relationship has grown apart, which may due to the decline of Jia Family贾府. But it's perhaps due to the second author's writing ability.
- JiaYuanchun, JiaYingchun, JiaTanchun and JiaXichun元迎探惜 are cousins. But Yuanchun is very far from others, because she is an imperial concubine. LiWan李纨 is not educated, but she is closer to Lin林 and Xue薛 than talented ShiXiangyun史湘云, which may be due to ShiXiangyun's late appearance. Qinwen晴雯 and Xiren袭人 are both impressive servant girls. But Xiren is closer to JiaBaoyu.
- JiaYucun贾雨村 is closer to others in the last 40 chapters than in the first 80 chapters, which may be not the CaoXueqin曹雪芹's idea. JiaYucun is close to LinZhixiao林之孝, which is easy to forget.

[2]Actually, the author of first 80 chapters is Cao Xueqin, which is relatively evident. But the claim that Gao E is the author of last 40 chapters is less evident.

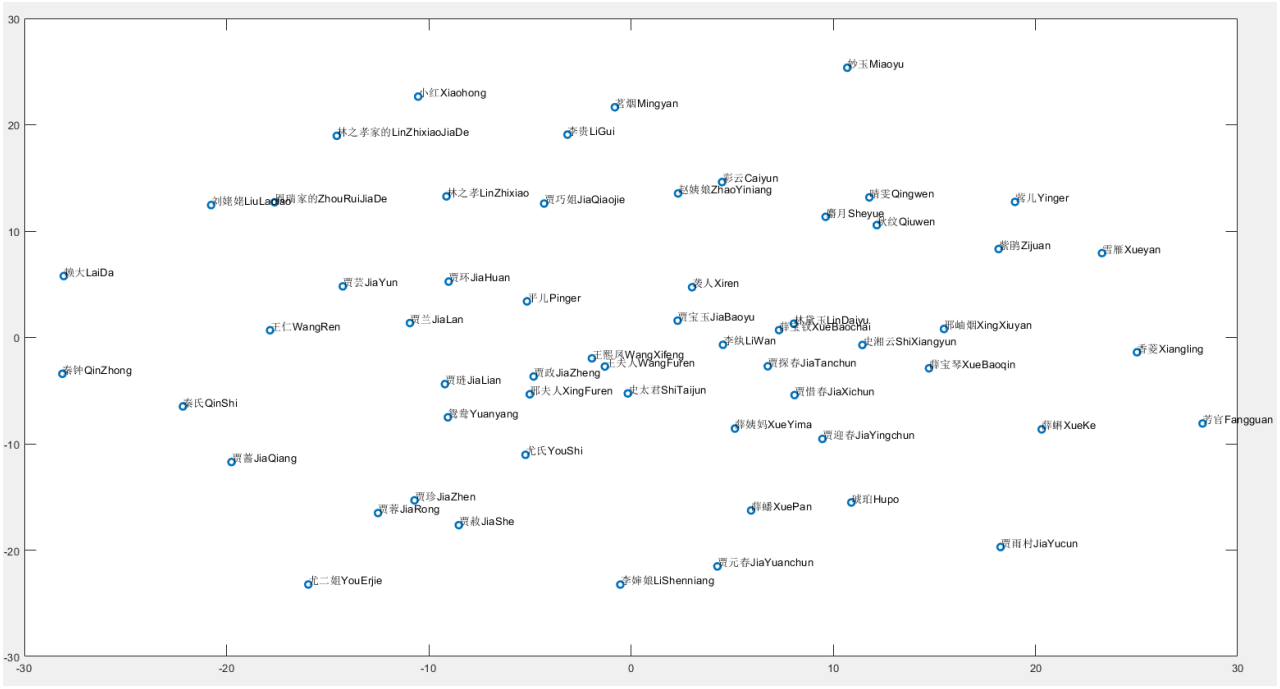

Figure 1: 120 Chapters (475 events)

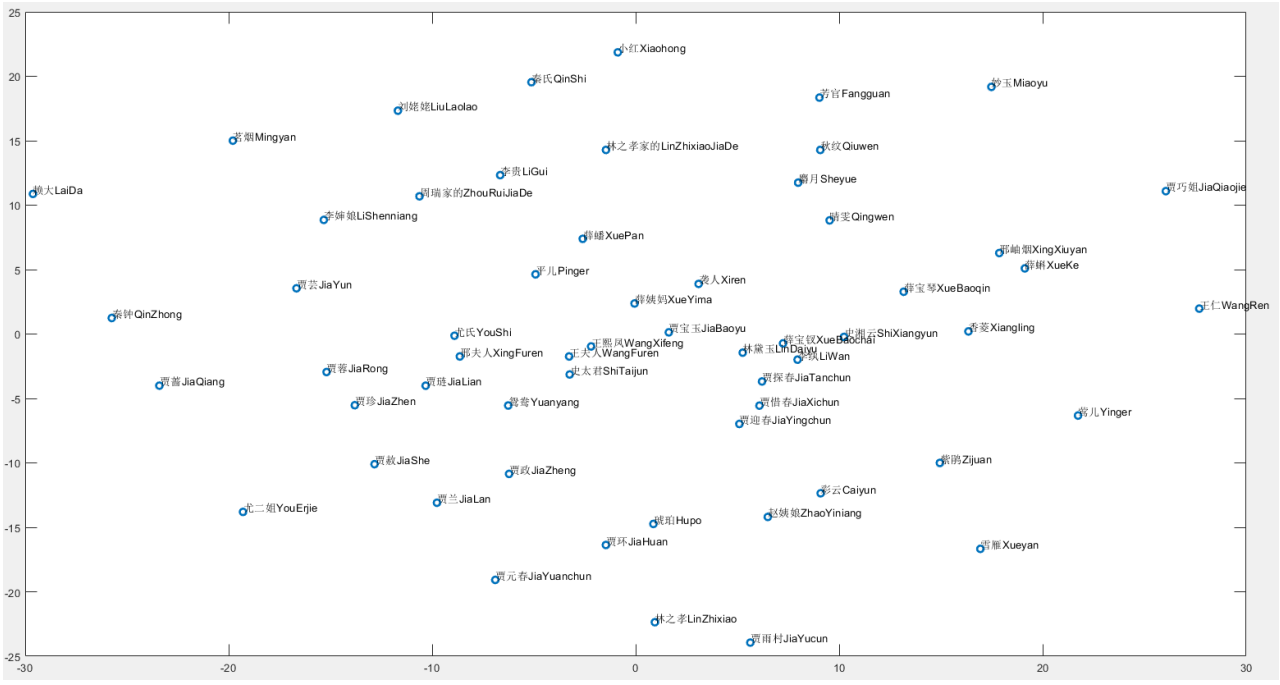

Figure 2: 80 Chapters (350 events)

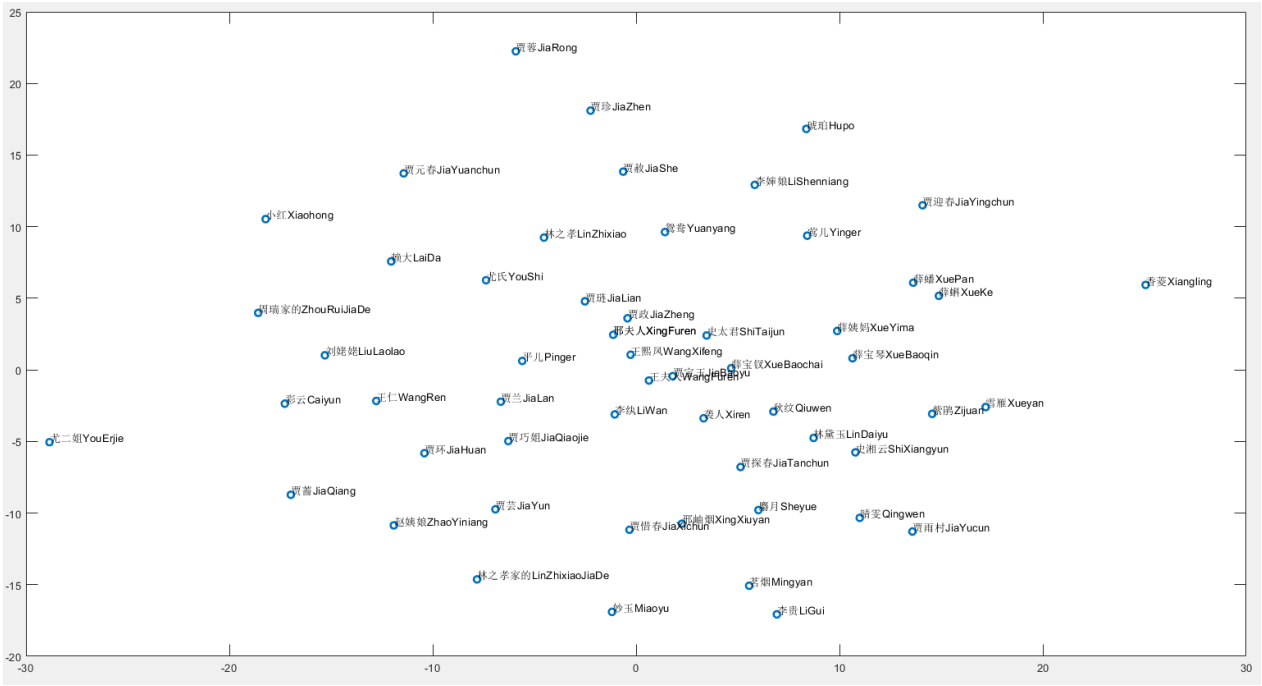

Figure 3: 40 Chapters (125 events)

4 Sparse Principal Component Analysis

In this section, we will use sparse principal component analysis to make a reduction of the data. We want to see who are the main roles in these two parts of the novel. And based on the result, we can find some difference between them.

4.1 Model Introduction

In this section, we use sparse PCA(principal component analysis) to evaluate the correlation between characters and events. From a dimension reduction perspective, PCA can be described as a set of orthogonal linear transformations of the original variables such that the transformed variables maintain the information contained in the original variables as much as possible.

Let $X \in \mathbb{R}^{n\times p}$ data matrix where n and p are the number of samples and variables, respectively. Preprocessing the data matrix, let $Y = X - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T X$, where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ and all its components composed by 1. We know, all column means of Y are 0. And the covariance matrix is calculated by $A = (Y^TY)/n$. Consider the following optimization programming,

$$\min_{x\in\mathbb{R}^p} -x^T\Sigma x, \ \text{ s.t. } \|x\| = 1. \tag{1}$$

This optimization problem is to find the eigenvector α_1 of A corresponding to the maximal eigenvalue. We can simply obtain the first principal component by calculating $Z_1 = X\alpha_1$. To better interpret the principal component, the author of [8] proposed a method named SPCA(sparse principal component analysis) and used ridge LASSO type method to find a 'sparse eigenvector'. In our report, we try to use an ℓ_0 constrained programming to find the sparse principal component as follow:

$$\min_{x\in\mathbb{R}^p} -x^TAx \ \text{ s.t. } \|x\| = 1, \|x\|_0 \le k, \tag{2}$$

We use Truncated Power Method proposed in [6] to solve the above problem. For the completeness of this report, we show the algorithm below. Firstly, we define a operation:

$$[\text{Truncate}(x, I)]_i = \begin{cases} [x]_i & i \in I \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 2: Truncated Power Method for solving 2

Initialization: $\Sigma \in \mathbb{R}^{p\times p}$, initial vector $x_0 \in \mathbb{R}^p, k = 1$.
Repeat:
Compute $x'_t = Ax_{k-1}/\|Ax_{t-1}\|$.
Let $I_t = \text{supp}(x'_t)$ be the indices of x_k with the largest k absolute values.
Compute $\widehat{x}_t = \text{Truncate}(x'_t, I_t)$.
Normalize $x_t = \widehat{x}_t/\|\widehat{x}_t\|$.
$t \leftarrow t + 1$.
Until Convergence.

It remains a problem that how we can compute the second sparse principal component. The author of [9] give us an answer. Suppose that x_1 is the first principal component obtained by Algorithm 2. Then we make a deflation on the covariance matrix A: $A_1 = A - (x_1^T Ax_1)x_1x_1^T$ and use Algorithm 2 again to compute the sparse principal component as the second sparse principal component of A. Based on this, we can start our numerical experience.

4.2 Numerical Experience

From the data introduction we know the first 350 events happened on $1 - 80$ chapters in the novel so we extract this 319×350 sub-matrix and record it as X_1. And the remain one is denoted by X_2 with 319 rows and 125 columns. We first make a preprocession on these two data matrix as stated in the last subsection. And we use Algorithm 2 to calculate the first two principal components of each covariance matrix. 10 largest absolute values of principal eigenvector and Solution of Algorithm 2 for 1-80 chapters

	principal eigenvector	k = 10	k=8	k=6	k=4	k=3	k=2	k=1
林黛玉Lin Daiyu	-0.5029	-0.5387	-0.5458	-0.5613	-0.6093	-0.6277	-0.5891	-1
贾宝玉Jia Baoyu	-0.4447	-0.4789	-0.5227	-0.5661	-0.5687	-0.6081	-0.8081	
薛宝钗Xue Baochai	-0.4391	-0.4585	-0.4572	-0.4576	-0.4907	-0.4860		
史湘云Shi Xiangyun	-0.2467	-0.2560	-0.2596	-0.2502	-0.2543			
贾探春Jia Tanchun	-0.2352	-0.2543	-0.2302	-0.1945				
李纨Li Wan	-0.1765	-0.1896	-0.1683					
袭人Xi Ren	-0.1950	-0.1891	-0.2099	-0.2336				
史太君Shi Taijun	-0.1591	-0.1676	-0.1645					
贾迎春Jia Yingchun	-0.1335	-0.1447						
贾惜春Jia Xichun	-0.1266	-0.1391						
Variance	0.4159	0.4161	0.4030	0.3883	0.3633	0.3452	0.3010	0.2041

4.3 Discussion of the experience results

We first give an explaination to Table 4. When set $k = 10$, the second sparse eigenvector is only of 7 cardinalities. We think the reason is that Yuan Yang, Feng Er AND Wang Ren have little contribution of the second principal component. So when we compute the sparse one, they are vanished. And the reason is similar in the case when $k = 8$.

When we focus on the first 80 chapters, the first 2 eigenvectors are still emotional line and political line, respectively. This phenomenon is consistent with [1].

We know from Table 4.2 and Table 3 that there is a gap between the main characters in 1-80 chapters and that of 81-120 chapters. For example, in Chapter 1-80, according to the results of sparse principal component analysis, Lin Daiyu is the most important role. But in the 81-120 rounds, her contribution only ranks seventh,which is due to her early death. And Xi Ren plays an important role in 81-120 chapters while she only ranks seventh in the first 1-80 chapters. For the second sparse eigenvalue of covariance matrix, Wang Xifeng, Wang Furen and Xing Furen has great contribution in both parts. But for other characters, there is a rather big difference in these two parts of the novel. JIaLian's

Table 2: 10 largest absolute values of second eigenvector and second sparse eigenvector for 1-80 chapters

	second eigenvector	k = 10	k=8	k=6	k=4	k=3	k=2	k=1
王熙凤Wang Xifeng	0.4620	0.4975	0.5185	0.5407	0.5751	0.6056	0.7801	1
史太君Shi Taijun	0.4409	0.4944	0.5012	0.5487	0.5569	0.5806	0.6257	
王夫人Wang Furen	0.4073	0.4720	0.4766	0.5040	0.5262	0.5432		
邢夫人Xing Furen	0.2691	0.2825	0.2888	0.2873	0.2867			
贾珍Jia Zhen	0.2172	0.2096	0.2210	0.1824				
尤氏You Shi	0.1993	0.1990	0.2033	0.1918				
贾蓉Jia Rong	0.1975	0.1898	0.2031					
贾琏Jia Lian	0.1952	0.1865	0.1948					
薛姨妈Xue Yima	0.1407	0.1668						
鸳鸯Yuan Yang	0.1361	0.1527						
Variance	0.3858	0.3869	0.3709	0.3500	0.3326	0.3105	0.2463	0.1951

Table 3: 10 largest absolute values of principal eigenvector and Solution of Algorithm 2 for 81-120 chapters

	principal eigenvector	k = 10	k=8	k=6	k=4	k=3	k=2	k=1
贾宝玉Jia Baoyu	0.4084	0.5054	0.5454	0.6030	0.6835	0.7512	0.7968	1
袭人Xi Ren	0.3552	0.4248	0.4513	0.4731	0.4832	0.5657	0.6042	
王夫人Wang Furen	0.3426	0.3409	0.3094	0.3089	0.3884	0.3402		
史湘云Shi Xiangyun	0.3268	0.3300	0.3268	0.3137	0.3854			
薛宝钗Xue Baochai	0.3019	0.3703	0.3943	0.3943				
紫鹃Zi Juan	0.2232	0.2645	0.2643	0.2346				
林黛玉Lin Daiyu	0.2063	0.1889	0.1900					
薛姨妈Xue Yima	0.2003	0.1976	0.1965					
秋纹Qiu Wen	0.1873	0.1831						
邢夫人Xing Furen	0.1807	0.1536						
Variance	0.5662	0.4823	0.4603	0.4358	0.3738	0.3444	0.3274	0.2064

Table 4: 10 largest absolute values of second eigenvector and second sparse eigenvector for 81-120 chapters

	second eigenvector	k = 10	k=8	k=6	k=4	k=3	k=2	k=1
王熙凤Wang Xifeng	0.4471	0.5054	0.4983	0.5321	0.7215	0.7362	0.7465	1
贾琏Jia Lian	0.3754	0.3997	0.4070	0.5405	0.5569	0.6251	0.6654	
贾政Jia Zheng	0.3332	0.3808	0.3889	0.3860	0.3046	0.2593		
王夫人Wang Furen	0.3244	0.3869	0.4077	0.4305	0.3075			
邢夫人Xing Furen	0.2172	0.3009	0.3455	0.3436				
平儿Ping Er	0.2646	0.2554	0.2390	0.3018				
史湘云Shi Xiangyun	0.1799	0.2360	0.2525					
鸳鸯Yuan Yang	0.0896							
丰儿Feng Er	0.0864							
王仁Wang Ren	0.0862							
Variance	0.4587	0.4127	0.3895	0.3500	0.3227	0.3010	0.2651	0.2204

position is much higher in the last 40 chapters than in the first 80 chapters. JiaLian's identity is key in the big family, but in Cao Xueqin's setting, his position is less important because of his personality. And actually, his wife WangXifeng's ending is really different for the hint(凡鸟偏从末世来，都知爱慕此生才。一从二令三人木，哭向金陵事更哀。) in Chapter 5. This detail also shows the

difference ideas between authors. But analyzing the two plot lines of the two parts, we can find that they always revolve around Jia Baoyu and Wang Xifeng in narration.

5 Conclusion

We applied non-Metic MDS and Sparse PCA in characters of *Dream of the Red Chamber*. The diagram from MDS is able to show some interesting relations and the tables for SPCA can show the main characters of two parts of the novel.In addition to objectively reflecting character relationships, these results also explain the several points mentioned above that many readers have overlooked. And our results also reflect the difference between the first 80 chapters and the last 40 chapters. These results may be not helpful to literature scholars, but they are interesting examples of data science at least.

6 Remark

Ma Chiyu mainly completed the third section, and Wu Yuqia did the fourth section.

References

[1] Mengting Wan. Character-Event Analysis and Network Analysis in *Dream of the Red Chamber*. Bachelor thesis, Yuanpei College, Peking University,2013.

[2] Shepard, Roger N. "The analysis of proximities: multidimensional scaling with an unknown distance function. I." Psychometrika 27.2 (1962): 125-140.

[3] Shepard, Roger N. "The analysis of proximities: Multidimensional scaling with an unknown distance function. II." Psychometrika 27.3 (1962): 219-246.

[4] Kruskal, Joseph B. "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis." Psychometrika 29.1 (1964): 1-27.

[5] Kruskal, Joseph B. "Nonmetric multidimensional scaling: a numerical method." Psychometrika 29.2 (1964): 115-129.

[6] Xiaotong Yuan and Tong Zhang. "Truncated Power Method for Sparse Eigenvalue Problems." Journal of Machine Learning Research 14(2013) 899-925.

[7] Hui Zou and Lingzhou Xue. "A Selective Overview of Sparse Principal Component Analysis." Proceedings of the IEEE, 2018, 106(8): 1311-1320.

[8] Hui Zou,Trevor Hastie, and Robert Tibshirani. "Sparse principal component analysis." Journal of computational and graphical statistics, 2006, 15(2): 265-286.

[9] Lester Mackey. "Deflation Methods for Sparse PCA." NIPS. 2008, 21: 1017-1024.