Handling missing values with Matrix completion

Multivariate analysis & Statistical learning

Leonardo Livi

Università degli Studi di Firenze

December 13, 2023

Matrix completion and Missing data

Often datasets have missing values, which can be a nuisance. How we can handle this issue?

- We could remove the rows that contain missing observations and perform our data analysis on the complete rows. But this seems wasteful, and depending on the fraction missing, unrealistic
- Alternatively, if x_{ij} is missing, then we could replace it by the mean of the jth column (using the non-missing entries to compute the mean). Although this is a common and convenient strategy.

Often we can do better by exploiting the correlation between the variables...

Matrix completion and Missing data (2)

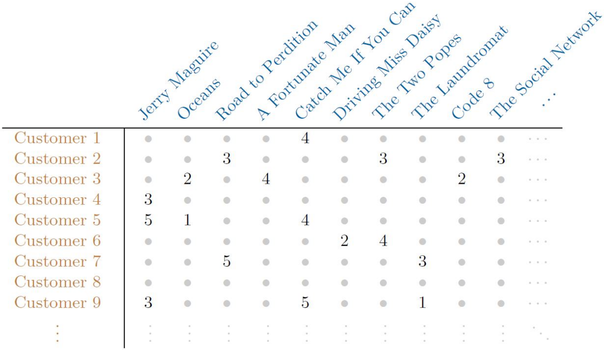
In this section we show how principal components can be used to impute the missing values, through a process known as matrix completion. The completed matrix can then be used in a statistical learning method, such as linear regression or LDA.

Remark

This approach for imputing missing data is appropriate if the missingness is random.

Recommender Systems

If we form a matrix of the ratings (on a scale from 1 to 5) that *n* customers have given to the entire Netflix catalog of *p* movies, then most of the matrix will be missing, since no customer will have seen and rated more than a tiny fraction of the catalog. If we can impute the missing values well, then we will have an idea of what each customer will think of movies they have not yet seen. Hence matrix completion can be used to power recommender systems.



	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	⋯
Customer 1	•	•	•	•	4	•	•	•	•	•	⋯
Customer 2	•	•	3	•	•	•	3	•	•	3	⋯
Customer 3	•	2	•	4	•	•	•	•	2	•	⋯
Customer 4	3	•	•	•	•	•	•	•	•	•	⋯
Customer 5	5	1	•	•	4	•	•	•	•	•	⋯
Customer 6	•	•	•	•	•	2	4	•	•	•	⋯
Customer 7	•	•	5	•	•	•	•	3	•	•	⋯
Customer 8	•	•	•	•	•	•	•	•	•	•	⋯
Customer 9	3	•	•	•	5	•	•	1	•	•	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

PCA with missing values

We now show how one can both impute the missing values and solve the principal component problem at the same time. We return to a modified form of the optimization problem in PCA setting:

$$\text{minimize}_{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}\left\{\sum_{(i,j)\in\mathcal{O}}\left(x_{ij}-\sum_{m=1}^{M}a_{im}b_{jm}\right)^{2}\right\}$$

where $\mathcal{O}$ is the set of all observed pairs of indices (i, j), a subset of the possible $n \times p$ pairs.

Hard-Impute algorithm

1. Create a complete matrix $\tilde{\boldsymbol{X}}$ of dimension $n \times p$ of which the (i, j) elements equals:

$$\tilde{x}_{ij} = \begin{cases} x_{ij} \text{ if } (i,j) \in \mathcal{O} \\ \bar{x}_j \text{ if } (i,j) \notin \mathcal{O} \end{cases}$$

2. Repeat steps (a)-(c) until the objective fails to decrease:
   (a) Solve

$$\text{minimize}_{\mathbf{A} \in \mathbb{R}^{n \times M}, \mathbf{B} \in \mathbb{R}^{p \times M}} \left\{ \sum_{j=1}^{p} \sum_{i=1}^{n} \left(\tilde{x}_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2 \right\}$$

   by computing the PC of $\tilde{\boldsymbol{X}}$

   (b) For each element $(i, j) \notin \mathcal{O}$, set $\tilde{x}_{ij} \leftarrow \sum_{m=1}^{M} \hat{a}_{im} \hat{b}_{jm}$

   (c) Compute the objective

$$\sum_{(i,j) \in \mathcal{O}} (x_{ij} - \sum_{m=1}^{M} \hat{a}_{im} \hat{b}_{jm})^2$$

3. Return the estimated missing entries $\tilde{x}_{ij}, (i, j) \notin \mathcal{O}$

Clash Royale dataset

Clash Royale is a strategic mobile game where players build their own army, collect cards featuring powerful characters and spells, and battle against opponents in real-time. It combines elements of card collecting, tower defense, and multiplayer battles.

Data description

This dataset contains information on all 107 cards of Clash Royale, including troops, buildings and spells. The information contained in this dataset include:

- Card name
- **Cost**
- **Count**
- **Damage**
- **Damage per second**
- Death damage
- **Hitpoints**
- **Hit speed**
- **Range**
- Type
- Rarity
- Evolution
- Ability
- **Win rate %**
- **Rating**
- **Usage %**

Not of all variables are included in the analysis

Some descriptive

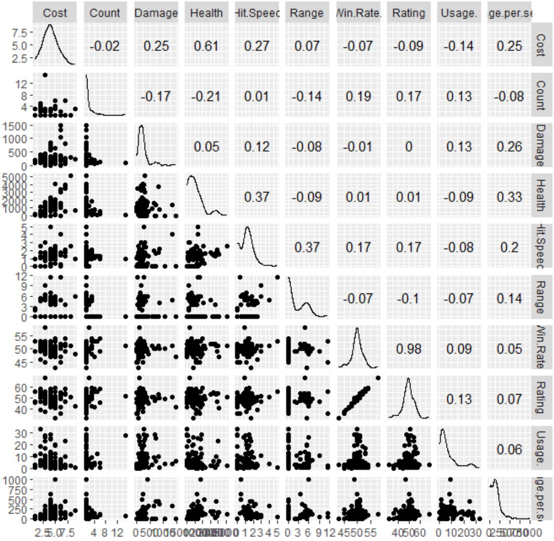


Figure: Scatterplot matrix

Missing randomness

There are $p = 10$ variables and $n = 107$ observation (cards). We first standardized the data so each variable has mean zero and standard deviation one. We then selected each observation and set one of the ten variables to be missing. Thus, 10% of the elements of the data matrix were missing.

Choice of M by simulation

We must select $\boldsymbol{M}$, the number of principal components to use for the imputation. We therefore proceed by simulation to choose the best M and then return to the implementation of the algorithm →

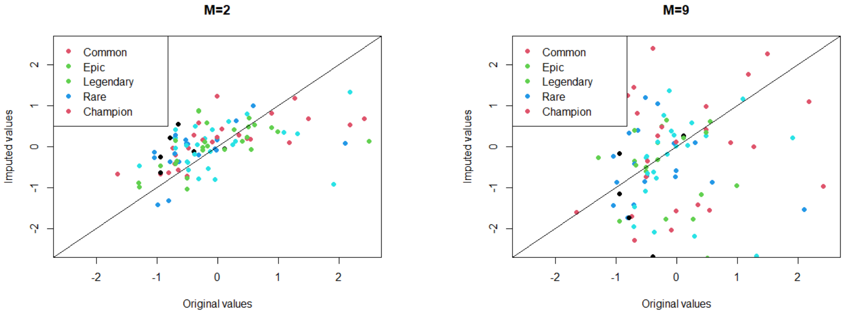
M	**Correlation (ρ)**
1	0.125
2	0.376
3	0.337
4	0.236
5	0.297
6	0.139
7	0.269
8	0.297
9	0.373

Hard Impute application

We can now apply the Hard Impute algorithm in pratice:

- Approximate the matrix with missing entries X^{NA} using the first M principal component by *singular values decomposition*
- Replace each missing value with the estimate from the algorithm and we will obtain $\hat{X}$
- Calculate the mean sum square as performance metric: $MSS = \frac{1}{n} \sum_{(i,j) \in \mathcal{O}} (x_{ij}^{NA} - \hat{x}_{ij})^2$
- Compute the relative error: $RE = \frac{MSS - MSS_{old}}{MSS_0}$ where MSS_{old} is the MSS calculate in the previous iteration and MSS_0 is the quadratic mean of the original matrix X^{NA}

True vs. imputed values plot



Evaluation of performance

For which value of M do we have the best performance?
To answer this question, we can compare this correlations ($\hat{\rho}$) to what we would have gotten if we had estimated these 107 values using the complete data (ρ):

	M = 2	**M = 9**
Iteration	18	1164
$\hat{\rho}$	0.376	0.373
ρ	0.629	0.99
$\Delta_{\rho,\hat{\rho}}$	0.253	0.617

We can notice the difference between the two implementation

Conclusion

- It is reasonable to assume that the algorithm with 9 principal components reports a correlation close to 1, since the missing value imputed as a linear combination of components a and b using all principal components is exactly that value.
- Although the estimates of performance measures (correlations) are almost identical for both cases, the algorithm with two principal components converges faster and the performance measure is closer to the true one

References

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning*, Second Edition, Corretted Printing: June 21, 2023.