Matrix Completion

Hamidreza Behjoo [1] **Rahim Tariverdi** [1] **Jaspers W. Huanay Quispe** [1]

Abstract

In this paper three well known algorithms for matrix completion are discussed. In the first place these algorithms are applied for missing pixel recovery in image, and their performance in compared. In the second place the best algorithm in terms of accuracy and speed is selected to build a movie recommender system.

1. Problem Statement

Usually we want to recover a rank-r matrix of $\mathrm{M} \in \mathbb{R}^{m \times n}$ of which we have only measured only a fraction of its entries. Generally speaking this problem is not feasible, but if we add some extra assumption on M, we can solve the problem. To solve the problem we need mn measurement, but if the rank is small , there is hope, as the number of degrees of freedom is $r(m+n-r) \ll mn$.

One possible solution is to find the lowest rank matrix which satisfies our measurement. That is,

$$\begin{array}{ll} \text{minimize} & \operatorname{rank}(\mathrm{X}) \\ \text{subject to} & X_{ij} = M_{ij},\ (i,j) \in \Omega \end{array} \tag{1}$$

In general, however, this optimization problem is a challenging nonconvex optimization problem which is NP-hard to solve and requires worst-case exponential running time in both theory and practice.

1.1. Convex Relaxation

One possible way to solve 1 is to relax rank with a convex one Candes & Tao, 2010.

$$\begin{array}{ll} \text{minimize} & \|\mathrm{X}\|_* \\ \text{subject to} & X_{ij} = M_{ij},\ (i,j) \in \Omega \end{array} \tag{2}$$

where $\|\mathrm{X}\|_* = \sum_i \sigma_i(\mathrm{X})$.

[1] Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Hamidreza Behjoo <Hamidreza Behjoo@skoltech.ru>.

Final Projects of the Machine Learning 2020 Course, Skoltech, Moscow, Russian Federation, 2020.

1.2. Singular Value Thresholding

The nuclear norm optimization problem for matrix completion can be efficiently addressed by using the singular value thresholding (SVT) algorithm (Cai et al., 2010), which is a first-order algorithm approximating the nuclear norm optimization problem by

$$\begin{array}{ll} \text{minimize} & \tau\|\mathrm{X}\|_* + \frac{1}{2}\|\mathrm{X}\|_F^2 \\ \text{subject to} & X_{ij} = M_{ij},\ (i,j) \in \Omega \end{array} \tag{3}$$

where τ is the parameter, for large values of τ the model is too low rank and is not able to fit the data, so both the training and test error is high. When τ is too small, the model is not low rank, which results in over fitting: the observed entries are approximated by a high-rank model that is not able to predict the test entries.

The algorithm start with an initial matrix $\mathrm{Y}^{(0)}$, where $\mathrm{Y}^{(0)}_{ij} = M_{ij}$ for $(i,j) \in \Omega$ and $\mathrm{Y}^{(0)}_{ij} = 0 \notin \Omega$, SVT applies an iterative gradient descent algorithm such that

$$\begin{aligned} &\mathrm{X} = D_\tau(\mathrm{Y}^{(i)}) \\ &\mathrm{Y}^{(i+1)} = \mathrm{Y}^{(i)} + \delta P_\Omega(\mathrm{M} - \mathrm{X}^{(i)}) \end{aligned} \tag{4}$$

where δ is the step size, P_Ω is an orthogonal projector onto Ω and D_τ knows as SVT operator. Given $\mathrm{Y}^{(i)}$ at ith SVT iteration step and its singular value decomposition (SVD) $\mathrm{Y}^{(i)} = \mathrm{U}^{(i)}\Sigma^{(i)}\mathrm{V}^{(i)^T}$ where $\mathrm{U}^{(i)}$ and $\mathrm{V}^{(i)}$ are orthogonal matrix and $\Sigma^{(i)} = diag(\sigma_1^{(i)}, \sigma_2^{(i)}, ..., \sigma_r^{(i)})$ is a diagonal matrix with $\sigma_1^{(i)} \geq \sigma_2^{(i)} \geq ..., \sigma_r^{(i)} \geq 0$ as the singular values of $\mathrm{Y}^{(i)}$, the SVT operator $D_\tau(\mathrm{Y}^{(i)})$ is defined as shrinking the singular values less than τ well as their associated singular vectors.

$$D_\tau(\mathrm{Y}^{(i)}) = \sum_j^{\sigma_j \geq \tau} (\sigma_j^{(i)} - \tau) u_j^{(i)} v_j^{(i)^T} \tag{5}$$

Computing $D_\tau(\mathrm{Y}^{(i)})$ is the main operation in SVT, which is required to be repeatedly carried out at every iteration. A straightforward way to estimate $D_\tau(\mathrm{Y}^{(i)})$ is to compute full SVD on $\mathrm{Y}^{(i)}$ and then shrink the small singular values below threshold.

1.3. Robust PCA

Principal component analysis (PCA) plays a crucial role in the analysis of high-dimensional data (Vaswani et al., 2018) and is a widely used dimensionality reduction technique (Xu et al., 2009). It involves solving a low-rank approximation which can be easily computed for moderately sized problems by computing the singular value decomposition (SVD). Over the last decade PCA has been extended to allow for missing data (matrix completion) or data with either corrupted or few entries inconsistent with a low-rank model(robust PCA).

Robust PCA (RPCA) solves a low-rank plus sparse matrix approximation, with the sparse component allowing for few but arbitrarily large corruptions in the low-rank structure; that is, a matrix $\mathrm{M} \in \mathbb{R}^{m\times n}$ is decomposed into a low-rank matrix L plus a sparse matrix S,

$$\min_{\mathrm{X}\in\mathbb{R}^{m\times n}} \|\mathrm{X}-\mathrm{M}\|_F \quad \text{s.t.} \quad \mathrm{X} \in \mathrm{LS}_{m,n}(r,s) \tag{6}$$

where $\mathrm{LS}_{m,n}(r,s)$ is the set of $m \times n$ matrices that can be expressed as a rank r matrix L plus sparsity s matrix S,

$$\mathrm{LS}_{m,n}(r,s) = \{\mathrm{L}{+}\mathrm{S} \in \mathbb{R}^{m\times n} : \mathrm{rank}(\mathrm{L}) \le r,\ \|\mathrm{S}\|_0 \le s\}$$

Solving RPCA as formulated in 6 is an NP-hard problem in general. Provable solutions for the problem were first provided in (Chandrasekaran et al., 2011; Candes & Tao, 2010) by solving the convex relaxation of the problem

$$\min_{\mathrm{L}\in\mathrm{R}^{m\times n}} \|\mathrm{L}\|_* + \lambda\|\mathrm{S}\|_1 \quad \text{s.t.} \quad \mathrm{M} = \mathrm{L}+\mathrm{S} \tag{7}$$

where $\|.\|_*$ denotes nuclear norm of a matrix. and $\|.\|_*$ denotes the l_1 norm of a vectorized matrix (sum of absolute values of its entries).

RPCA is closely related to the problem of recovering a low-rank matrix from incomplete observations, referred to as matrix completion (Recht et al., 2010). The main difference between the two is that in the case of a matrix completion, the indices of missing entries are known, and the aim is to solve

$$\begin{aligned} &\min_{\mathrm{L}\in\mathbb{R}^{m\times n}} \|P_\Omega(\mathrm{L}) - P_\Omega(\mathrm{M})\|_F \\ &\text{s.t.} \quad \mathrm{L} \in \mathrm{LS}_{m,n}(r,0), |\Omega_c| = s, \end{aligned} \tag{8}$$

To facilitate fast and efficient solution, we use a family of algorithms called Augmented Lagrange Multiplier (ALM) methods (Lin et al., 2010), shown to be effective on problems involving nuclear norm minimization. Augmented Lagrange is defined as follows

$$\begin{aligned} l(\mathrm{L},\mathrm{S},\mathrm{Y}) = \|\mathrm{L}\|_* + \lambda\|\mathrm{S}\|_1 + \mathrm{tr}\{\mathrm{Y}^T(\mathrm{M}-\mathrm{L}-\mathrm{S})\} \\ +\frac{\mu}{2}\|\mathrm{M}-\mathrm{L}-\mathrm{S}\|_F^2, \end{aligned}$$

the details of the method is discussed in (Candes et al., 2009). we only mention here our choice of μ and λ

$$\mu = \frac{nm}{4\|\mathrm{M}\|_1}, \quad \lambda = \frac{1}{\sqrt{\max(n,m)}}, \tag{9}$$

which are usually selected in practice.

2. Experiments

In this section we do experiments with introduced algorithms in previous section. We choose two tangible application among many different applications; missing pixels recovery in images and movie recommender system.

2.1. Image Experiment

Due to the limitation of computational resources, we focus on gray-scale images, extending to the color images are straightforward. Matrix completion algorithms tries to find a low rank representation of the missing data, so before moving further its wise to check is fulfilled. In Figure 1 the image and its singular value distribution are shown. Only the first 50 singular values are significant and the rest are very small and can be discarded, so the image satisfies low rankness.

In figure 2 the first row show the images with different levels of salt and pepper noise, while the other rows shows

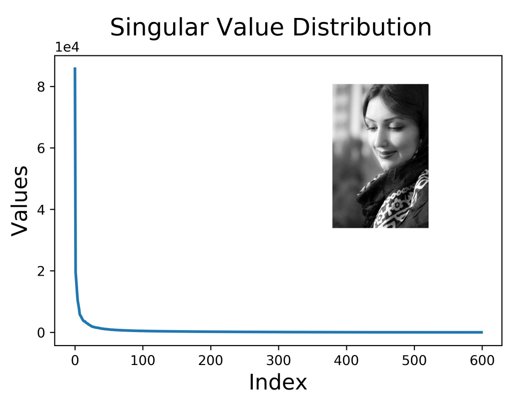

Figure 1. Singular Value Distribution of the test image

Figure 2. Images with different level of missing pixels and their corresponding recovery algorithms

recovered images with their corresponding algorithm. Here CVX (Diamond & Boyd, 2016) is the name of convex optimization package that we used to minimize the nuclear norm.

In order to compare different algorithm quantitatively we use Structural Similarity Index (SSIM) (Zhou Wang et al., 2004) as a metric. The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing the image. It is a full reference metric that requires two images from the same image capture—a reference image and a processed image. Fortunately in Python scikit-image there is a built-in function to calculate SSIM. SSIM is widely used by image processing community and is a well-known criterion for image quality assessment.

In order for the algorithms to have a good performance to recover the image, we need to tune their corresponding parameters. In the case of CVX the only hyper parameter is μ which controls trade off between nuclear norm and square loss. We always set this parameter to 1 as it gives the best performance. In the case of RPCA the hyper parameters are chosen based on equation 9 which works fine for practical purposes. Figure 3 shows how SVT works, every line correspondence to different ratio of noise starting from 10% to 60%. To show the effect of each parameter on SVT's performance for image recovery we do a grid search over different value of parameters. Parameters which are included in the study are top k largest singular values (svt_k) and learning parameter (svt_delta). Talking more specifically about svt_k, for a specific k, let's say 10, we can see the algorithm has slightly lost its efficiency. From the different point of view, for a chosen value of svt_delta, not all but in most cases we see that the larger value of k, the better performance of algorithm. Also as we can expect, the algorithm behaves differently for different ratios of noise added to the original picture.

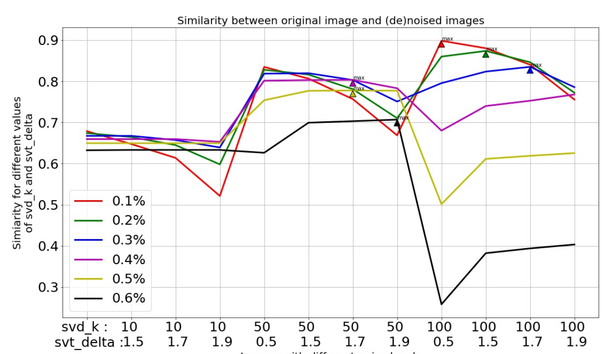

Figure 3. Parametric study of SVT algorithm

After parametric study of different algorithm, we apply each algorithm with best hyper parameter on images in figure 1. The results of CVX (second row) and SVT (third row) are comparable and they are able to recover the images with 60% of missing pixels (salt and pepper noise), while RPCA (fourth row) performance is acceptable up to 30% of missing pixels. To better compare these algorithms the SSIM for each one of them is plotted in 4.

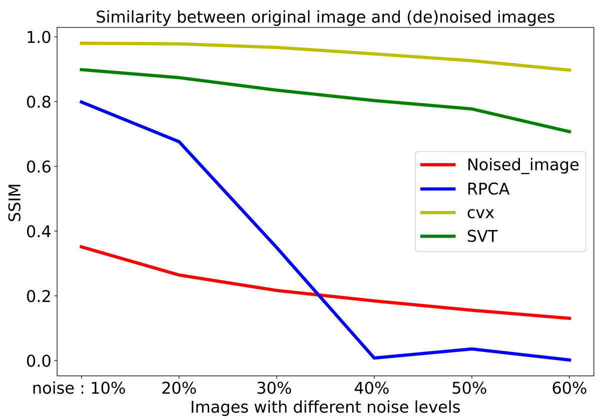

Figure 4. SSIM of different algorithm vs noise level

Based on this figure we can draw some conclusions as follows:

- SVT performance is quite good and comparable to CVX, and can recover the image even when 60% of the pixels are lost. SVT is quite fast in comparison to CVX and its implementation is strighforward.
- CVX performance is quite good and can recover the image up to 60 % noise level. Although it is a good algorithm to recover missing pixels, it is computationally expensive and takes a lot of time for even this small size problem.
- RPCA performance start to degrade when the noise level increases. It has good performance when the noise is sparse.

So we conclude that SVT is the winner in terms of accuracy and speed. We pursue the rest of paper with SVT algorithm and use it to build a recommender system.

2.2. Movie Recommender System

A recommender system, or a recommendation system (sometimes replacing 'system' with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. As a proof of the importance of recommender systems, we can mention that, a few years ago, Netflix organised a challenges (the "Netflix prize") where the goal was to produce a recommender system that performs better than its own algorithm with a prize of 1 million dollars to win.

In this paper we use **MovieLens** for doing our experimnet. This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

Table 1. Top 15 movies recommended for user 1

User 1 Recommendation	
Movie Name	Rating
The Yards (2000)	5.2810
Up in Smoke (1978)	5.2301
Lawrence of Arabia (1962)	5.2236
The Princess Bride (1987)	5.1427
Arsenic and Old Lace (1944)	5.1302
Some Kind of Wonderful (1987)	5.1139
A Walk on the Moon(1999)	5.1086
The Wizard of Oz (1939)	5.1084
Dumb and Dumber (1994)	5.0728
Tombstone (1993)	5.0620
Home Alone 2 (1992)	5.0614
Dirty Dancing (1987)	5.0567
The French Connection (1971)	5.0545
A Fish Called Wanda (1988)	5.0497
Fargo (1996)	5.0439

Before applying SVT on MovieLens dataset, we should do some prepossessing on dataset to convert them in an appropriate form an an input for the algorithm. The details of preprocessing procedure are in the Jupyter Notebook. Our goal is to predict rating for those movies that the user has not been rated. We split the dataset and take 10% for test and 90% for training. Root Mean Square (RMS) error for prediction of SVT algorithm on test samples is 16%, which is a good accuracy in the case of recommender systems. The result of 15 best prediction for user 1 is listed in table 1. The output of our algorithm is not integer numbers and can be seen in table 1.

3. Conclusion

In this paper we examine different algorithms for matrix completion and apply them on missing pixels recovery in images. We see that in terms of accuracy and computational burden SVT outperform CVX and RPCA, then we apply SVT on MovieLens dataset to predict unrated movies by user and we obtained an RMS about 16% on test samples which the great potential of matrix completion in recommender system.

4. Future Work

In every iteration of SVT we need to calculate a full SVD, which is really time consuming and will become a big challenge in large scale problem. Randomized SVD decomposition can be used to accelerate every iteration and this method can be applied on a very large scale dataset to see its performance.

5. GitHub

All the source code and everything related to this project can be found in the following link: https://github.com/jwilliamn/Matrix-Completion.

References

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010. doi: 10.1137/080738970. URL https://doi.org/10.1137/080738970.

Candes, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Candes, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis?, 2009.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011. doi: 10.1137/090761793. URL https://doi.org/10.1137/09076179.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Lin, Z., Chen, M., and Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, 2010.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010. doi: 10.1137/070697835. URL https://doi.org/10.1137/070697835.

Vaswani, N., Chi, Y., and Bouwmans, T. Rethinking pca for modern data sets: Theory, algorithms, and applications [scanning the issue]. *Proceedings of the IEEE*, 106(8): 1274–1276, 2018.

Xu, H., Caramanis, C., and Mannor, S. High dimensional principal component analysis with contaminated data. In *2009 IEEE Information Theory Workshop on Networking and Information Theory*, pp. 246–250, 2009.

Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Hamidreza Behjoo (35% of work)

- Writing the report
- Writing theoretical part of presentation
- Reviewing literate for matrix completion (3 papers)
- Writing Code for RPCA and doing experiment on images
- Coding the main algorithm
- Distributing task among team members

Rahim Tariverdi (33% of work)

- Reviewing literate for matrix completion (3 papers)
- Writing Code for SVT and doing experiment on images and movie recommender
- Coding the main algorithm
- Preparing the presentation

Jaspers W. Huanay Quispe (32% of work)

- Reviewing literate for convex relaxation applied to matrix completion
- Writing code for convex relaxation and doing experiment on images
- Github setup of the project
- Preparing slides for presentation

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   General comment: In the case of convex optimization we used the cvxpy library, but it requires to formulate the problem in an appropriate way. In the case of SVT there is a well implemented code by Candes in MATLAB, we used that code and translate it to python. In the case of RPCA there was a readily available code in python and we changed it to adapt to our own problem, In this case 40% the code was written by our team member. **Students' comment:** None

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   - [ ] Yes.
   - [ ] No.
   - [x] Not applicable.

   Students' comment: The pre-processing step in the case of MovieLens dataset are done in Jupyter Notebook.

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: In the case of recommender system we used 10% of sample for test, which we state it in the report.

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: We did a grid search study in the case of SVT to tune the parameters.

9. The exact number of evaluation runs is included.

   - [ ] Yes.
   - [ ] No.
   - [x] Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: we plot SSIM which is an indicator of error for every algorithm.

13. A description of the computing infrastructure used is included in the report.

    - [ ] Yes.
    - [ ] No.
    - [x] Not applicable.

    Students' comment: None