Exercises
Computational Intelligence Lab
SS 2020

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Thomas Hofmann
Web http://da.inf.ethz.ch/cil

Series 2, Feb 27/28, 2020 (SVD)

A recommender system is concerned with suggesting items (e.g. books on Amazon, movies at Movielens or music at lastFM) that are likely to interest the user. In collaborative filtering, we base our recommendations on the (known) preference of the user towards other items, and also take into account the preferences of other users.

One naive approach to collaborative filtering is to apply SVD to the training data matrix, completed with zeros at all unobserved entries. The data comes as a matrix $\mathbf{A}$, where each row represents an item and each column represents a user. Each entry $\mathbf{A}_{i,j}$ indicates the rating of item i by user j.

Problem 1 (SVD Theory):

Viewers were asked to rate a list of movies (on a scale of 1-10). The results are presented in the following table, the * sign indicates a missing values (the user did not watch the movie).

	Ben	Tom	John	Fred	Jack
American pie	8	7	1	*	4
Shrek	9	7	2	5	6
Titanic	1	4	9	*	3
The godfather	3	*	8	5	4
Avatar	*	3	*	9	9
Star wars	5	1	4	10	*

We can think of this rating table as a matrix $\mathbf{A}$. In our case, $\mathbf{A} \in \mathbb{N}^{6,5}$.

1. Consider the matrix $\mathbf{K} = \mathbf{A}\mathbf{A}^T$. What is the size of this matrix? What does $\mathbf{K}_{ij}$ tell us?
2. Consider the matrix $\mathbf{L} = \mathbf{A}^T\mathbf{A}$. What is the size of this matrix? What does $\mathbf{L}_{ij}$ tell us?
3. In SVD, we decompose $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. What should be the sizes of $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{V}$? (answer without looking at the matrices below).

   Here we used an SVD implementation to compute the decomposition. To do so we replaced each missing item with an average rating of 5.5

$$\mathbf{A} = \underbrace{\begin{pmatrix} -0.38 & -0.51 & 0.23 & -0.42 & 0.60 & 0.02 \\ -0.43 & -0.51 & 0.23 & 0.26 & -0.61 & -0.24 \\ -0.33 & 0.58 & 0.34 & -0.19 & 0.04 & -0.63 \\ -0.37 & 0.34 & 0.45 & 0.06 & -0.11 & 0.73 \\ -0.50 & 0.11 & -0.43 & 0.64 & 0.38 & -0.04 \\ -0.41 & 0.11 & -0.63 & -0.55 & -0.32 & 0.13 \end{pmatrix}}_{\mathbf{U}} \underbrace{\begin{pmatrix} 29.7 & 0 & 0 & 0 & 0 \\ 0 & 10.00 & 0 & 0 & 0 \\ 0 & 0 & 7.09 & 0 & 0 \\ 0 & 0 & 0 & 2.75 & 0 \\ 0 & 0 & 0 & 0 & 0.67 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{D}} \underbrace{\begin{pmatrix} -0.44 & -0.60 & 0.02 & -0.11 & -0.66 \\ -0.37 & -0.25 & 0.73 & -0.07 & 0.51 \\ -0.39 & 0.74 & 0.35 & 0.06 & -0.41 \\ -0.56 & 0.16 & -0.50 & -0.56 & 0.31 \\ -0.45 & -0.04 & -0.30 & 0.81 & 0.20 \end{pmatrix}^T}_{\mathbf{V}}$$

4. What can we hope to gain by performing SVD on the rating matrix?
5. Consider the entries in $\mathbf{D}$, how many singular values is it reasonable for us to keep? Which ones would you keep, why?
6. What interpretation can you give for the matrix $\mathbf{U}$?
7. What interpretation can you give for the matrix $\mathbf{V}$?

We now want to reduce the dimensionality of our data to $k = 2$ by keeping only the first two singular values.

8. Plot the two dimensional representations of the movies onto a two-dimensional grid and give a possible interpretation of the two axes.

   Hint: There are multiple ways to achieve this. One way is the following: Note that since $\mathbf{V}$ is orthonormal, we have $\mathbf{V}^\intercal\mathbf{V} = \mathbf{V}\mathbf{V}^\intercal = \mathbf{I}$ and hence

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\intercal \Leftrightarrow \mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{D}.$$

   As a result you can either plot the first two columns of $\mathbf{AV}$ or $\mathbf{UD}$ which in both cases contains the embedding of movie i in the $i - th$ row (e.g. $(\mathbf{m}_1^\intercal\mathbf{u}_1, \mathbf{m}_1^\intercal\mathbf{u}_2)$).

9. Plot the two dimensional representations of the users on a two-dimensional grid and give a possible interpretation of the two axes. (*Hint:* Just like for 8. there are multiple ways to acomplish this)

10. Observing the plots (and applying some cinema knowledge), how do you interpret the matrix $\mathbf{D}$?

11. Write down the approximation of the original matrix $\mathbf{A}$ using only the first three singular components.

12. What would be the approximation error under the euclidean and Frobenius norm?

13. Next, Bob wants to join the system. He shares a few of his movie ratings ([1,*,*,6,*,10]). How can we use the system to recommend movies for Bob?

    (a) Represent Bob in the new coordinate system and add this data to the plot.

    (b) Find the most suitable recommendation for Bob. (There is more than one way to do this.)

14. Does Bob's rating affect our prediction system?

Problem 2 (SVD for Collaborative Filtering):

In this assignment, we will apply singular value decomposition (SVD) to build our own recommender system. The recommender systems task is one of the possible project tasks, and we present it first here.

Submission system environment setup:

1. The web page for the collaborative filtering competition can be found here. To join the competition, create a kaggle account using your `....ethz.ch` email address.

   https://inclass.kaggle.com/c/cil-collab-filtering-2019.

   (in order to join, you need the private invite link on the course homepage)

2. Download the provided dataset `data_train.csv`, as well as `sampleSubmission.csv`, from the competition webpage.

To submit your solution to the online evaluation system, we require you to prepare a ".csv" file of the same structure as `sampleSubmission.csv` (the order of the predictions does not matter, but you must use the same set of ids for the entries to predict).

Your submission is evaluated according to the root mean squared error of the predictions you submit.

Working with rating data: In the provided dataset, we observe that not all users rated all items, and hence, many ratings are not available. Therefore, we have to deal with *missing values*. In collaborative filtering, the goal is to predict missing values from the observed ones.

First draft: A simple baseline algorithm One very simple way to compute the missing values is to replace each missing value by the average rating for the respective item. We suggest starting with this simple solution, later we will investigate more sophisticated approaches.

Below is a solution pipeline with an evaluation step:

1. **Load Training Data:** Load the data matrix given in data_train.csv. This loads a matrix $\mathbf{X}$ of 1000 items by 10000 users.

   Each entry $\mathbf{X}_{i,j}$ contains a rating of item i by user j (between 1 and 5 stars). The goal is to predict all missing values using information only from the observed ones. The indices of the missing values we're interested in for the evaluation are provided in the file sampleSubmission.csv.

2. **Prediction:** In this simple baseline algorithm, we impute missing values in the matrix $\mathbf{X}^{pred}$ by setting them to the average over all observed ratings for a particular item, in $\mathbf{X}^{train}$. Since the matrices are of the same size, the index of an item (and user) in the train matrix is the same as in the predictions matrix.

3. **Evaluation:** Once you upload your solution to kaggle, your predicted values in $\mathbf{X}^{pred}$ will be compared to the true observed values in $\mathbf{X}^{test}$ (secret set) using the root mean squared error:

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\Omega}\left(\mathbf{X}_{i,j}^{pred} - \mathbf{X}_{i,j}^{test}\right)^2}{\sum_{(i,j)\in\Omega} 1}} \tag{1}$$

   where Ω is the subset of observed indices (i.e. those i, j pairs where we have to predict the ratings $\mathbf{X}_{i,j}^{test}$, as specified in sampleSubmission.csv).

Second draft: Improving prediction Using SVD In this part we assume that the data has an underlying structure where each user and each item has a representation in the same concept space. This kind of structure can be found using SVD. Performing SVD requires a full matrix, we can use the baseline solution as an initialization step to obtain predictions for the missing values. This initial step, although necessary, introduces noise which affects the SVD result. The hope is that the "true" underlying structure can be revealed by discarding the less informative part of the decomposition. The initial missing values can then be inferred using the approximated representation.

1. **Impute missing values:** Repeat previous steps.

2. **SVD decomposition:** Compute the SVD decomposition of the training matrix with imputed values:

$$\tilde{\mathbf{X}}^{train} = \mathbf{U}\mathbf{D}\mathbf{V}^\top.$$

   For simplicity, we recommend multiplying $\mathbf{U}$ and $\mathbf{V}$ by the square root of the eigenvalues:

$$\mathbf{U}' = \mathbf{U}\sqrt{\mathbf{D}},$$

$$\mathbf{V}' = \sqrt{\mathbf{D}}\mathbf{V}.$$

3. **Prediction:** The prediction for any missing value $\mathbf{X}_{i,j}$ can now be computed as the inner product of the i-th row in $\mathbf{U}'$ and the j-th column in $\mathbf{V}'^\top$.

4. **Model Selection:** Select a number k of eigenvalues to be used and truncate $\mathbf{U}$ and $\mathbf{V}$ accordingly. Compute predictions for test matrix using the truncated matrices and calculate the error (RMSE). Try different values of k, and observe what happens to your error rates.

5. **Recommender System:** Write a function to predict the (unknown) rating for a given index pair i, j.

Extensions: Naturally there are many ways to improve your SVD solution. More advanced techniques can be found in the following publications:

- Webb, B. (2006). Netflix Update: Try This at Home. Simon Funk's Personal Blog. http://sifter.org/~simon/journal/20061211.html

- Koren Y., Bell R., Volinsky B., "Matrix Factorization Techniques for Recommender Systems" IEEE Computer, Volume 42, Issue 8, p.30-37 (2009); http://research.yahoo.com/files/ieeecomputer.pdf

- A. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering," Proc. KDD Cup and Workshop, ACM Press, 2007, pp. 39-42.