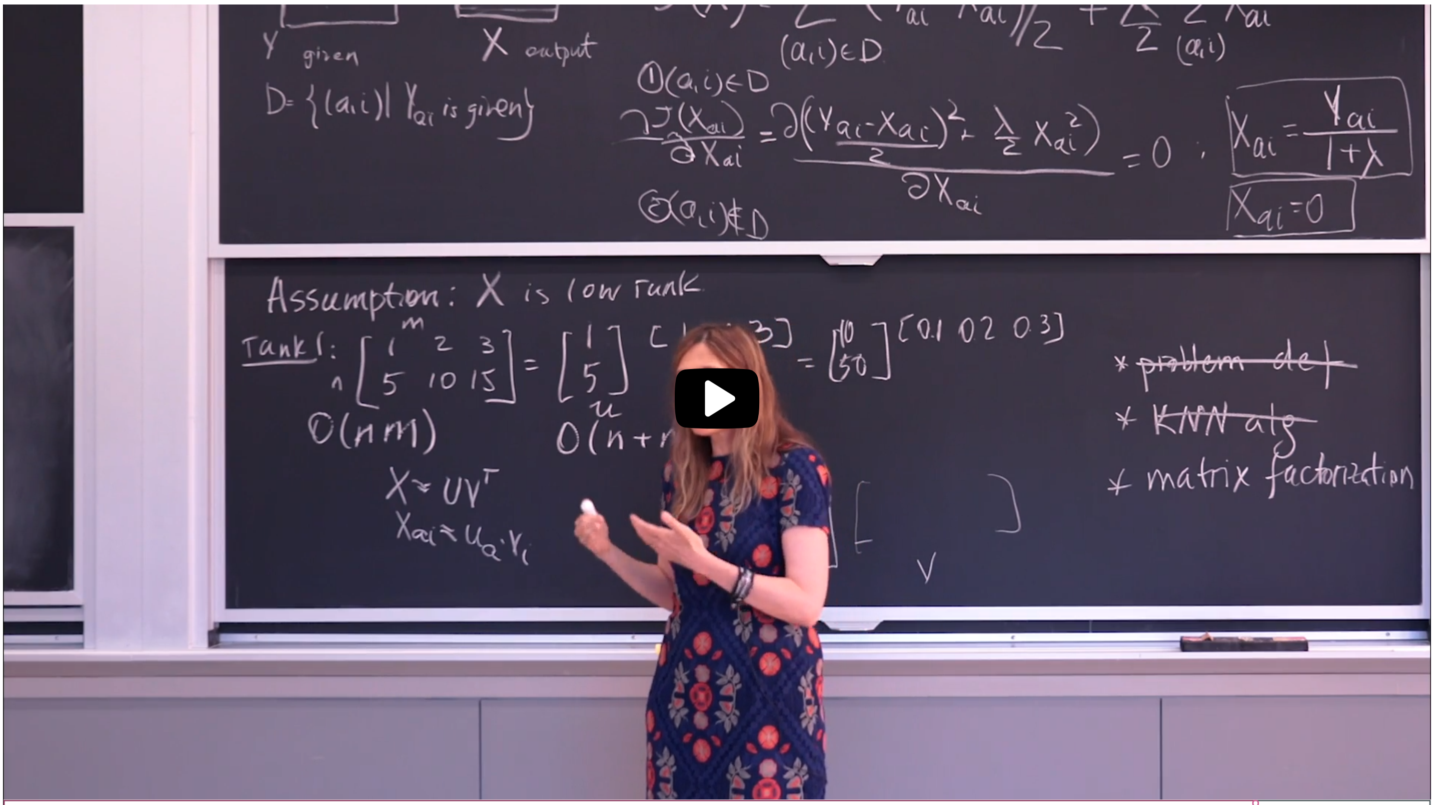
5. Collaborative Filtering with Matrix Factorization

Collaborative Filtering with Matrix Factorization



7:42 / 8:44 1.50x

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Matrix Factorization Practice

1/1 point (graded)

We now use **collaborative filtering** to solve the movie recommender system problem.

As we saw in the previous problem, we ended up with an unsatisfactory and trivial solution of X by minimizing the objective alone:

$$J(X) = \sum_{a,i \in D} \frac{(Y_{ai} - X_{ai})^2}{2} + \frac{\lambda}{2} \sum_{(a,i)} X_{ai}^2.$$

In the collaborative filtering approach, we impose an additional constraint on X:

$$X = UV^T$$

for some $n \times d$ matrix U and $d \times m$ matrix V^T. The number d is the **rank** of the matrix X.

Suppose

$$X = \begin{bmatrix} 3 & 6 & 3 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

then what is the minimum possible d?

$d =$ 1 ✔ **Answer:** 1

Solution:

X can be decomposed as

$$X = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Remark: Note that imposing that a n by m matrix X has rank $k < \min(m, n)$ means that some of its rows (*resp.* columns) are linearly dependent on other rows (*resp.* columns).

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Intuition on the Vector Factors

1/1 point (graded)

Assume we have a 3 by 2 matrix X i.e. we have 3 users and 2 movies. Also, X is given by

$$X = \begin{bmatrix} \text{User 1's rating on movie 1} & \text{User 1's rating on movie 2} \\ \text{User 2's rating on movie 1} & \text{User 2's rating on movie 2} \\ \text{User 3's rating on movie 1} & \text{User 3's rating on movie 2} \end{bmatrix} = UV^T$$

for some $3 \times d$ matrix U and $d \times 2$ matrix V^T.

Now which of the following is true about U and V^T? (Choose all those apply.)

- [x] The first row of U represents information on user 1's rating tendency
- [ ] The first row of U represents information on movie 1
- [ ] The first column of V^T represents information on user 1's rating tendency
- [x] The first column of V^T represents information on movie 1

✔

Solution:

U encodes information about the users, and V about the movies.

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering (2 weeks):Lecture 7. Recommender Systems / 5. Collaborative Filtering with Matrix Factorization

Add a Post

❮ All Posts

Excellent introduction to concepts behind singular-value decomposition!

discussion posted 8 months ago by **dkhachatrian**

Pinned

The thought process behind this matrix factorization of X into U and V^T is probably one of the clearest explanations I've seen for what the components of singular value decomposition (SVD) are. There are just a few differences between what we discussed in lecture and how SVD works:

1. In lecture, we assumed the $n \times m$ matrix X to be of some specific rank k. In SVD, we don't explicitly make that assumption (though we still have a way of determining the relative importance via the *singular values* -- more on that in a second).
2. In lecture, the professor mentioned how U and V^T can vary by scaling when writing $X = UV^T$. In SVD, we "capture" all of the scaling into a diagonal matrix Σ by writing $X = U\Sigma V^T$. The entries of Σ are the *singular values*, that describe the "strength of association" between the relevant entries in U and V^T. Small singular values mean that the association between the two are nearly non-existent -- singular values equal to zero mean that X is not full-rank.

Besides that, it's basically as was described in lecture, and the intuitive explanation given by the example was great! Recapitulating for the most part: If X_{ai} is what person a thought about movie i and we want to describe things with k "concepts", then

- the a'th row of $U\,(n \times k)$ describes Person a's relative affinity for k unknown/"latent" concepts, and
- the i'th column of $V^T\,(k \times m)$ describes how relevant these k latent concepts are to movie i.
- When using SVD, the $\Sigma_{j,j}$ entry of the matrix Σ describes: if Person a likes concept j and Movie i is highly relevant to concept j, how much effect does that actually have on the final rating X_{ai}?

Technically, what's described above is a *truncated SVD*. The *full SVD* is slightly hairier (and this extra hairiness is probably what makes SVD feel so opaque). For "full" SVD, you need to have Σ be an $n \times m$ rectangular-diagonal matrix, and U and V^T change shape accordingly (and be made *orthogonal/unitary* matrices $U\,(n \times n)$ and $V^T\,(m \times m)$ -- i.e., again, making sure Σ captures all of the scaling). Why? The number of nonzero entries in Σ is the rank of X, which is between 0 and $\min(n, m)$, so this ensures you're "safe" regardless of the rank of X.

But if you *know* the rank of X to be k -- or *want* the matrix of rank-k that is "close" of X -- then you can

1. do the truncated SVD with Σ as a $k \times k$ matrix and (appropriate shapes for U and V^T), as described above. When the rank of X is in fact $\leq k$, this gives the "same" result as full SVD. When $\operatorname{rank}(X) > k$, I believe this gives the "closest" rank-k matrix to X, but it does so by "mixing in" the less important "concepts" into the k concepts we have kept. Or;
2. do the full SVD, reorder the matrices so that the largest k singular values are at the top, and drop the rest of the rows/columns. This may be slightly less "close" compared to (1), but it doesn't perform any "extra" mixing like (1) does.

Which one you prefer would probably depend on the sizes of n, m, and k.

7/14/19 UPDATE

The good point made by @khanhedx in their response made me want to emphasize something:

SVD only works on matrices with *filled-in data*. Trying to perform SVD on Y directly (which is mostly filled with unknown entries) will cause errors or give erroneous results (if your marker for "unknown data" is a number, e.g., -1).

Also, ***different assumptions are made*** when imputing X via truncated SVD ($X = U\Sigma V^T$, rows of U and columns of V^T have unit norm) compared to $X = UV^T$. In SVD, as mentioned before, *all* scaling is captured by Σ and shared across all rows of U (people) and columns of V^T (movies), which all have *unit norm*. What this means is, for example, there is no way for an X factorized according to the truncated SVD to capture the idea that, say, "Person a rates all types of movies more highly in general compared to Customer b" (mathematically, $U_{aj} > U_{bj}, j \in \{1, 2, \cdots, k\}$).

One could regain this capability by relaxing the restriction of unit norms -- at which point you can drop Σ altogether and recover the matrix factorization $X = UV^T$ discussed in lecture. Or you could perhaps keep Σ and play with regularization terms to allow but punish deviations of U and V^T from unit norm ("you're allowed to say that some users just like all movies more than others, but I don't want you to use that reason willy-nilly").

The proposed changes to the objective function above suggest different assumptions about how you expect the data to be structured and/or how you want the data to be described. This should be a good reminder of just how important it is to specify your objective function properly.

This post is visible to everyone.

Add a Response

6 responses

KaiTian199

8 months ago

Thanks for this highly informational post!

+1

posted 8 months ago by **ScottBernstein** prof. Barzilay's lectures ♡

posted 8 months ago by **TheNewParadigm**

Add a comment

karenechu (Staff)
8 months ago - endorsed 8 months ago by **dkonomis** (Staff)

pinning for everyone.

Add a comment

JenniferVoitle
8 months ago

and in case it helps, here is the wonderful Gilbert Strang. https://www.youtube.com/watch?v=mBcLRGuAFUk

Add a comment

sharov_am
8 months ago

https://www.youtube.com/watch?v=P5mlg91as1c&list=PLaG0xtWHUxMq1kQ71ZipuTmL3BN562kEZ&index=28

Add a comment

khanhedx
8 months ago

Excellent post @dkhachatrian! One thing I would add is that for matrix factorization for recommender system, the original matrix is sparse, so one has to come up with ways to fill in the missing values in the matrix before doing the SVD.

One method to fill in missing values is to subtract every existing rating by the user's average rating, then from that new matrix, subtract every rating by the movie's rating. Then, any missing value can be filled as 0. This effectively assumes that any and all missing rating is "perfectly average", although in reality that is often not the case. I think difficulty in imputing missing values is one of the reasons why SVD is not often used, and methods that only minimize errors in predicting existing ratings such as the one in the lecture are the common methods.

I think you're giving credits to the wrong person. Guess it's almost her signature, but not every long and insightful post is Pat's. :-)

posted 8 months ago by **mrBB** (Community TA)

Ahaha I'll fix it. Thanks mrBB!

posted 8 months ago by **khanhedx**

I had taken the mistake as a compliment exactly due to @ptressel's usual MO haha!

Thanks for the kind words and the great point @khanhedx! I should perhaps emphasize that SVD is more of a descriptive tool for an already-existing (and without missing values) matrix than the best method to generate an imputed matrix. Even notwithstanding the "missing values" problem, it is more constrained than the matrix factorization described in lecture -- since we're imposing unit vectors in U and V and capturing all the scaling in Σ and Σ is "shared" for every row of U and column of V^T, there is no way for the truncated SVD to capture the notion of, say, "Person a rates movies more highly in general compared to Customer b" (mathematically, $U_{a,j} > U_{b,j}, j \in \{1, 2, \cdots, k\}$). Let me point to this post and add this caveat.

posted 8 months ago by **dkhachatrian**

Haha no problem. About your comment about SVD not recovering the "true" U and V due to the pesky scaling factor Σ, one method to mitigate it (that I've come across in the literature) is to multiple U and V by $\sqrt{\Sigma}$. In other words, the user-factor matrix will be $U\sqrt{\Sigma}$ and the movie-factor matrix will be $\sqrt{\Sigma}V$.

posted 8 months ago by **khanhedx**

Add a comment

RommelAlbertoRodriguezPerez
8 months ago

Truly a great post @dkhachatrian, thanks.

Add a comment

Showing all responses

Add a response:

Preview

Submit