Matrix Completion For Collaborative Filtering

Probability and Statistics for Data Science

Carlos Fernandez-Granda

These slides are based on the book Probability and Statistics for Data Science by Carlos Fernandez-Granda, available for purchase here. A free preprint, videos, code, slides and solutions to exercises are available at https://www.ps4ds.net

Goals

Explain how to perform collaborative filtering with low-rank models

Show how to fit these models to data

Apply them to predict movie ratings

Motivation

Datasets where each data point is associated to 2 entities

1. *Recommender systems*: Rating given to movie i by user j
2. *Computational genomics*: Expression level of gene i in cell j
3. *Weather forecasting*: Temperature in location i at time j

Collaborative filtering

$$D := \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} ? & ? & 5 & 4 \\ ? & 1 & 4 & ? \\ 4 & 5 & 2 & ? \\ ? & 4 & 2 & 1 \\ 4 & ? & 1 & 2 \\ 1 & 2 & ? & 5 \end{pmatrix} \end{array} \begin{array}{l} \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array}$$

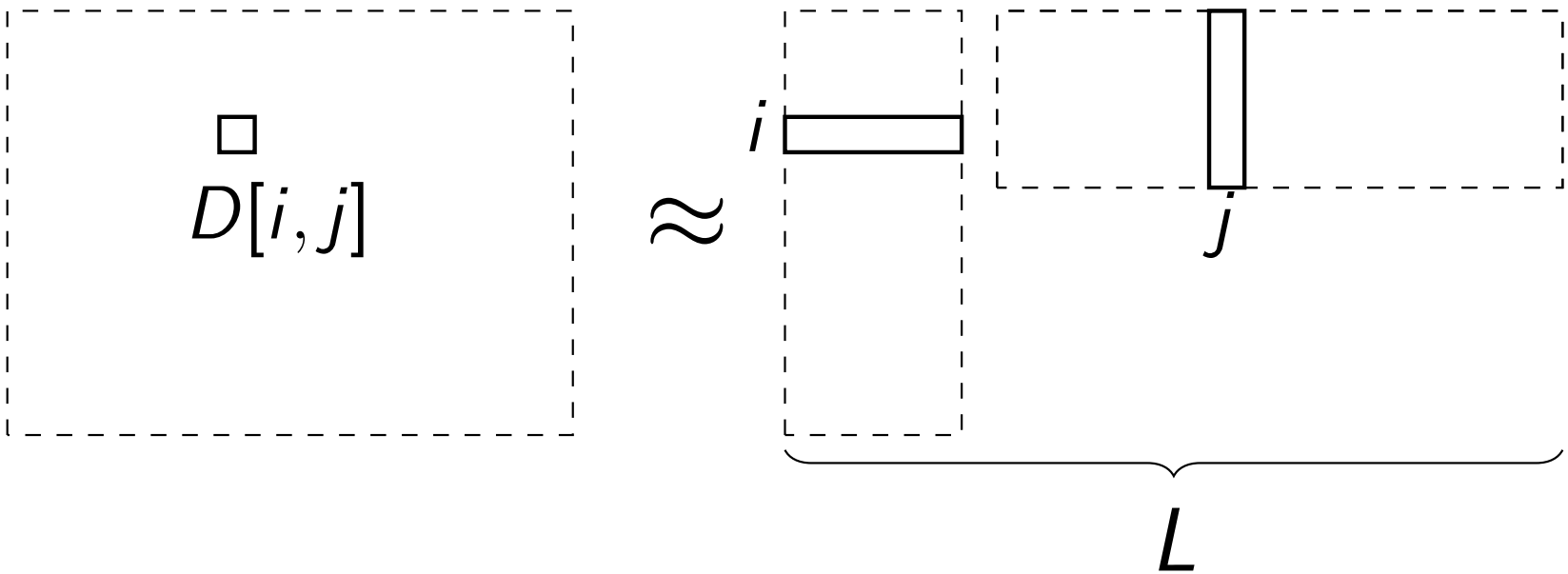
Matrix completion

$$\begin{bmatrix} 1 & ? & 1 \\ ? & 6 & 3 \\ ? & 4 & 2 \end{bmatrix}$$

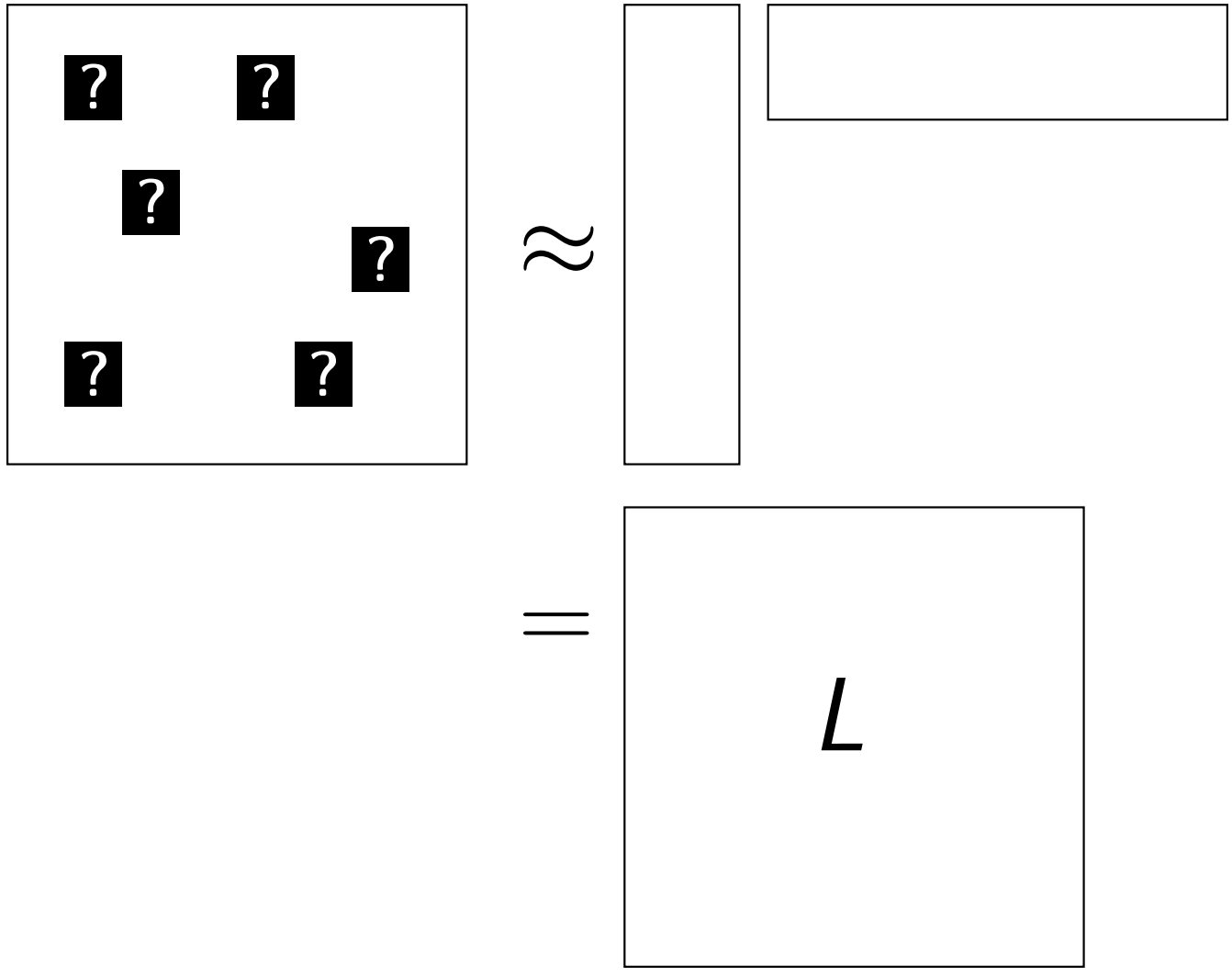
We can insert any values!

Low-rank model

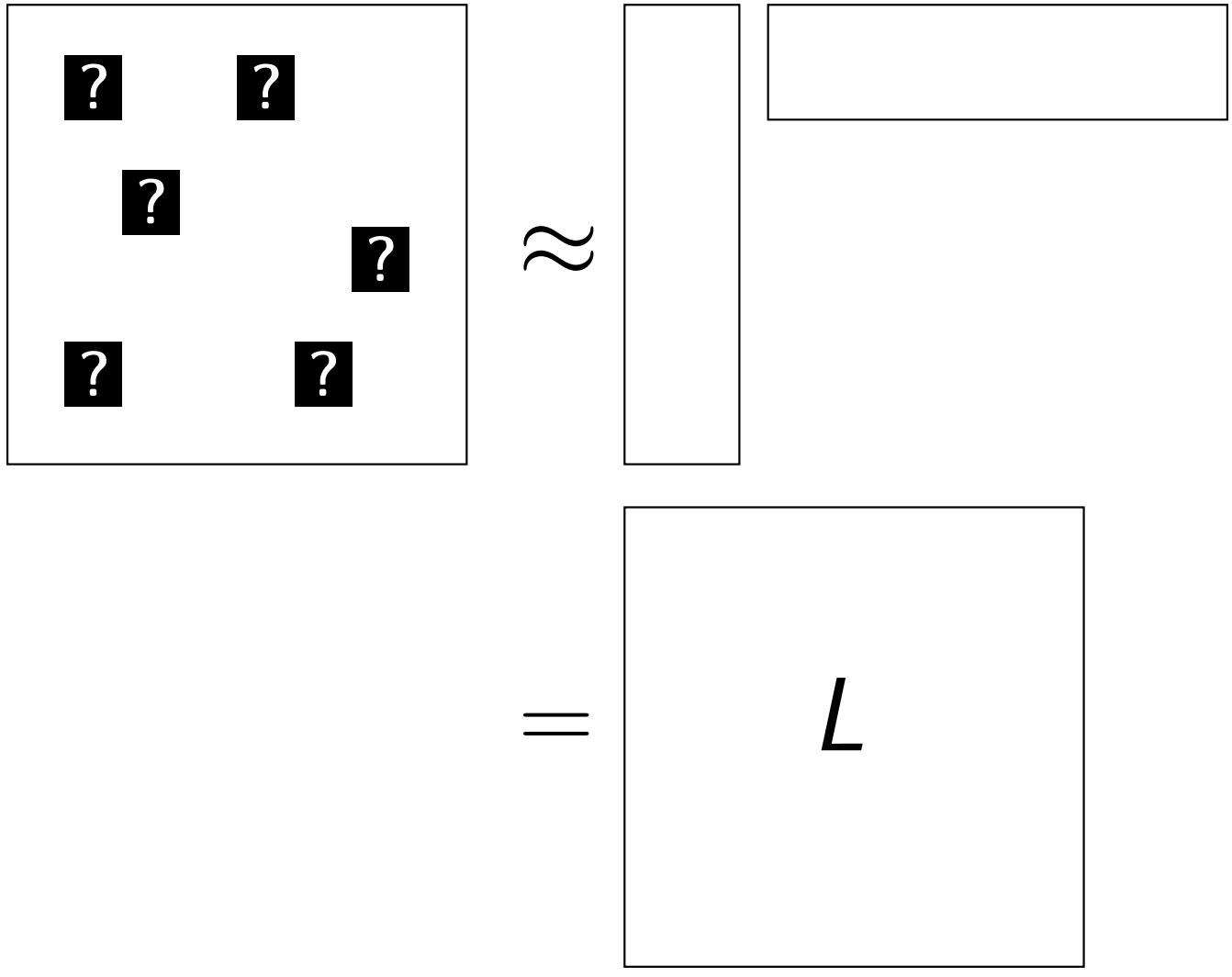
$$D[i,j] \approx L[i,j] := \sum_{l=1}^{r} a_l[i]\, b_l[j]$$



Low-rank matrix completion



Low-rank matrix completion



Low-rank matrix completion

Assumption: Rank $= 1$

$$\begin{bmatrix} 1 & ? & 1 \\ ? & 6 & 3 \\ ? & 4 & 2 \end{bmatrix} = ab$$

$$= \begin{bmatrix} a[1]\,b[1] & a[1]\,b[2] & a[1]\,b[3] \\ a[2]\,b[1] & a[2]\,b[2] & a[2]\,b[3] \\ a[3]\,b[1] & a[3]\,b[2] & a[3]\,b[3] \end{bmatrix}$$

For any α, $ab = \alpha a \frac{b}{\alpha}$, so we set $a[1] := 1$

Low-rank matrix completion

$$\begin{bmatrix} 1 & ? & 1 \\ ? & 6 & 3 \\ ? & 4 & 2 \end{bmatrix} = \begin{bmatrix} a[1]\,b[1] & a[1]\,b[2] & a[1]\,b[3] \\ a[2]\,b[1] & a[2]\,b[2] & a[2]\,b[3] \\ a[3]\,b[1] & a[3]\,b[2] & a[3]\,b[3] \end{bmatrix}$$

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

$$a[1] = 1 \qquad b[1] = a[1] = 1 \qquad b[3] = a[1] = 1$$

$$a[2] = \frac{3}{b[2]} = 3 \qquad b[2] = \frac{6}{a[2]} = 2 \qquad a[3] = \frac{4}{b[2]} = 2$$

Can we complete this matrix?

$$\begin{bmatrix} 1 & 1 & 1 \\ ? & ? & ? \\ 1 & 1 & 1 \end{bmatrix} = ab = \begin{bmatrix} 1 \\ ? \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

We need to observe every row and column

Can we complete this matrix?

True matrix:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 23 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Data:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & ? & 0 \\ 0 & 0 & 0 \end{bmatrix} = ab = \begin{bmatrix} 0 \\ ? \\ 0 \end{bmatrix} \begin{bmatrix} 0 & ? & 0 \end{bmatrix}$$

Only observed low-rank structure can be recovered

In practice

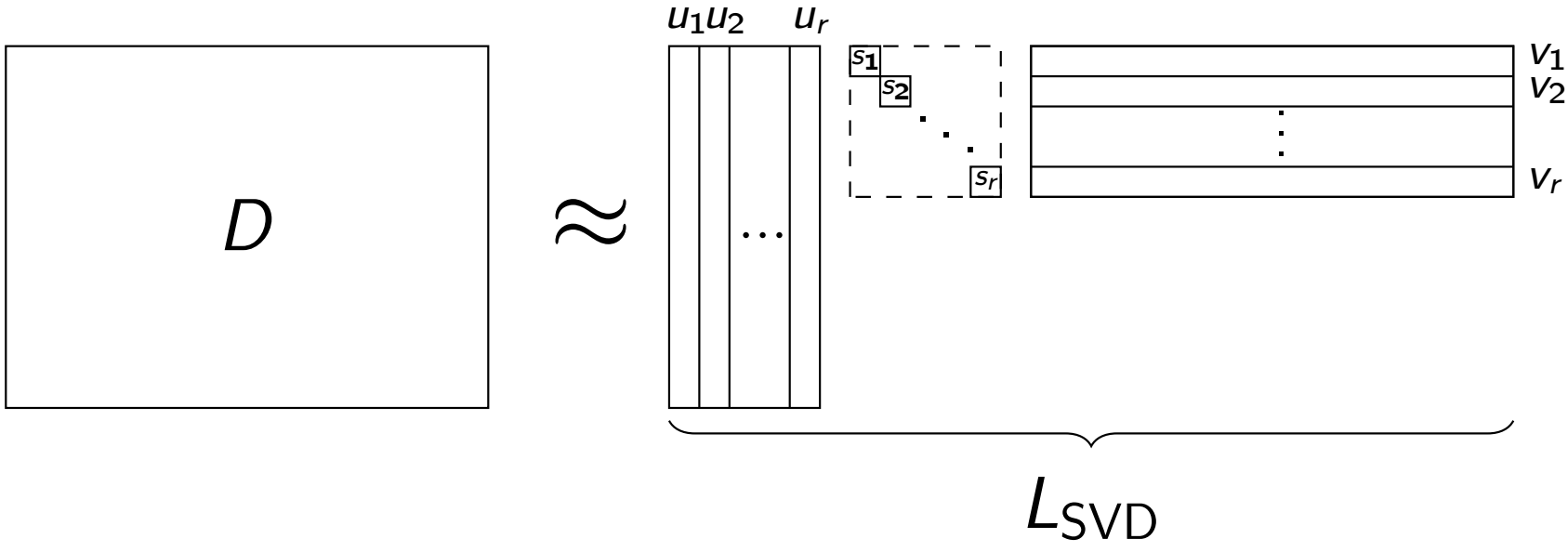
Data cannot be expected to be exactly low rank

Goal: Find low-rank matrix that is closest to the data

$$\sum_{(i,j)\in \text{observed}} \left(D[i,j] - \sum_{l=1}^{r} a_l[i] b_l[j] \right)^2$$

Problem: Nonconvex cost function that is difficult to optimize

Truncated SVD



Optimal if no entries are missing

$$L_{\text{SVD}} = \arg \min_{\text{rank}(L)=r} ||D - L||_{\text{F}}$$

Movie ratings

$$D := \begin{array}{c} \begin{array}{cccc} \text{Bob} & \text{Molly} & \text{Mary} & \text{Larry} \end{array} \\ \begin{pmatrix} ? & ? & 5 & 4 \\ ? & 1 & 4 & ? \\ 4 & 5 & 2 & ? \\ ? & 4 & 2 & 1 \\ 4 & ? & 1 & 2 \\ 1 & 2 & ? & 5 \end{pmatrix} \end{array} \begin{array}{l} \\ \text{The Dark Knight} \\ \text{Spiderman 3} \\ \text{Love Actually} \\ \text{Bridget Jones's Diary} \\ \text{Pretty Woman} \\ \text{Superman 2} \end{array}$$

How can we use the truncated SVD? Impute missing entries!

Centered imputed data

$$D_{\text{ct-imputed}} := \begin{bmatrix} 0 & 0 & 2.06 & 1.06 \\ 0 & -1.94 & 1.06 & 0 \\ 1.06 & 2.06 & -0.94 & 0 \\ 0 & 1.06 & -0.94 & -1.94 \\ 1.06 & 0 & -1.94 & -0.94 \\ -1.94 & -0.94 & 0 & 2.06 \end{bmatrix}$$

Singular values: $s_1 = 4.8 \quad s_2 = 2.48 \quad s_3 = 2.37 \quad s_4 = 1.46$

Full matrix: $s_1 = 7.79 \quad s_2 = 1.62 \quad s_3 = 1.55 \quad s_4 = 0.62$

Estimate ($m(D_{\text{imputed}}) = 2.94$)

$$L_{\text{initial}}[\text{movie}, \text{user}] := m(D_{\text{imputed}}) + s_1 u_1[\text{movie}] v_1[\text{user}]$$

Bob	Molly	Mary	Larry	
2.28 (1)	2.08 (1)	3.91 (5)	3.88 (4)	The Dark Knight
2.35 (2)	2.16 (1)	3.81 (4)	3.79 (5)	Spiderman 3
3.67 (4)	3.91 (5)	1.85 (2)	1.87 (1)	Love Actually
3.73 (5)	3.99 (4)	1.76 (2)	1.79 (1)	Bridget Jones's Diary
3.69 (4)	3.93 (5)	1.82 (1)	1.85 (2)	Pretty Woman
2.06 (1)	1.78 (2)	4.24 (5)	4.21 (5)	Superman 2

Are we happy with this?

Not completely happy...

Bad news: We are fitting the imputed values

Good news: The estimate is better than initial naive imputation

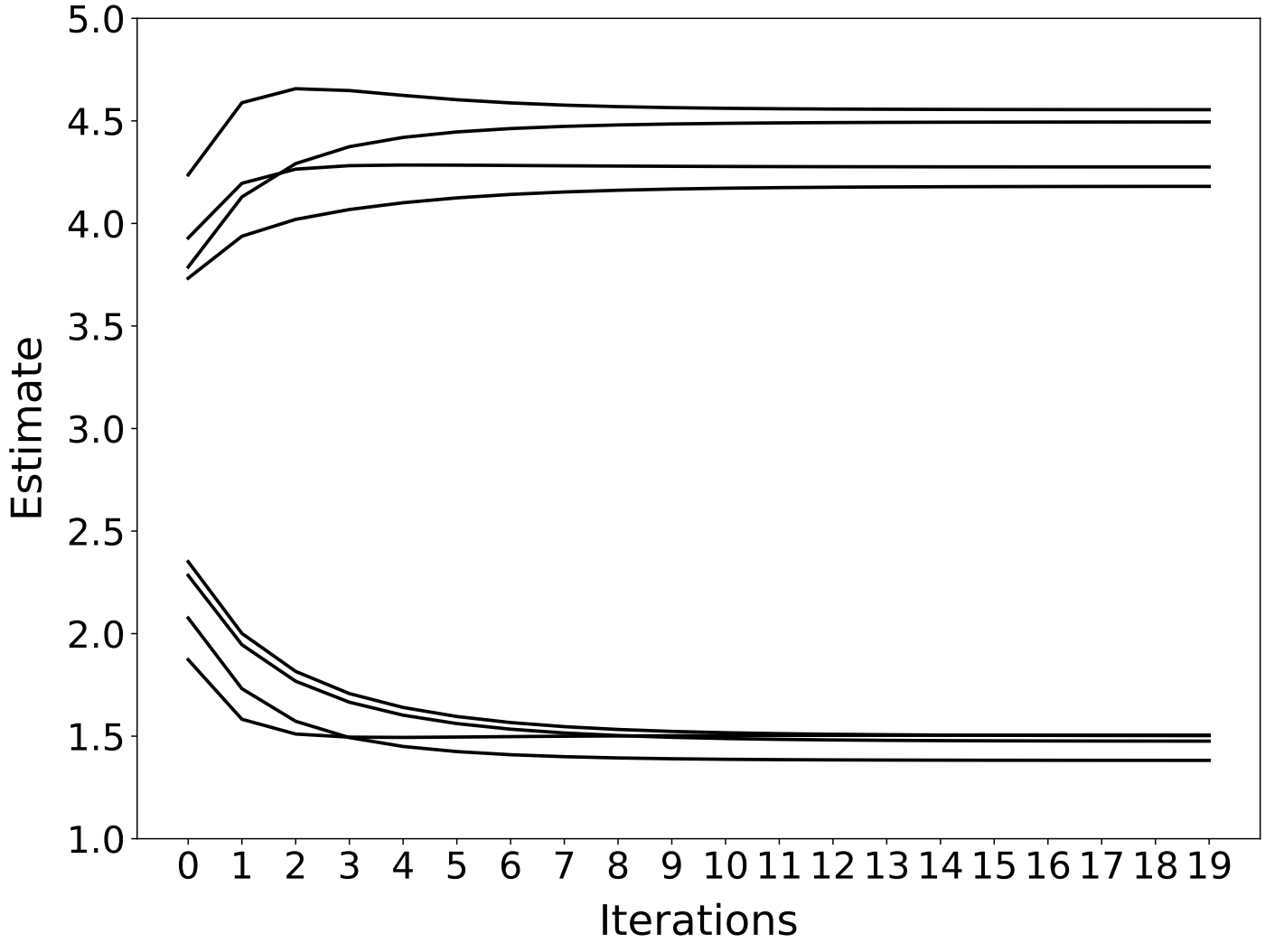
Idea: Alternate between imputation and SVD truncation

This method is called *hard-impute* [Mazumder et al 2009]

Other methods:

- Gradient descent [Keshavan et al 2009]
- Alternating least squares [Jain et al 2013]
- Convex relaxation [Candes, Recht 2009]

Missing entries ($m(D_{\text{imputed}}) = 2.94$)



Final estimate

$$L_{\text{final}}[\text{movie}, \text{user}] := m(D_{\text{imputed}}) + L_{\text{cnt}}[\text{movie}, \text{user}]$$

Bob	Molly	Mary	Larry	
1.48 (1)	1.38 (1)	4.45 (5)	4.52 (4)	The Dark Knight
1.50 (2)	1.41 (1)	4.42 (4)	4.50 (5)	Spiderman 3
4.26 (4)	4.34 (5)	1.57 (2)	1.51 (1)	Love Actually
4.18 (5)	4.26 (4)	1.65 (2)	1.59 (1)	Bridget Jones's Diary
4.2 (4)	4.28 (5)	1.64 (1)	1.57 (2)	Pretty Woman
1.37 (1)	1.27 (2)	4.55 (5)	4.63 (5)	Superman 2

Real data

Movielens dataset

1,000 users and 100 movies with most ratings

Observed ratings: 30,055 (69,945 are missing)

Validation set: 1,000 ratings

Test set: 1,000 ratings

Rank-r model

We impute missing entries using mean rating for each movie

We alternate between (1) truncating SVD and (2) re-imputing

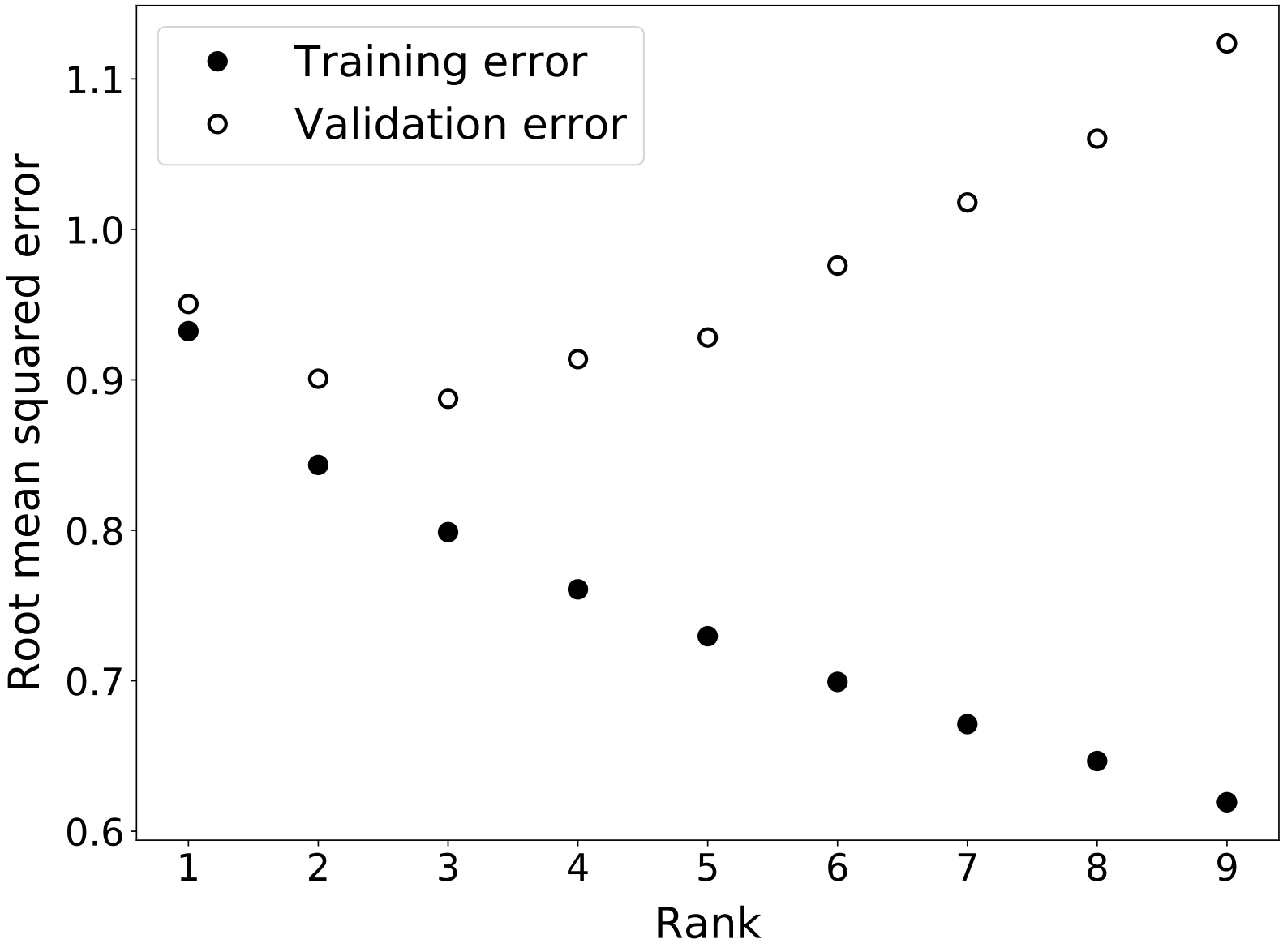
Important: How do we choose r?

Rank r

How do you expect r to affect:

- the training error?
- the validation error?

Training and validation error



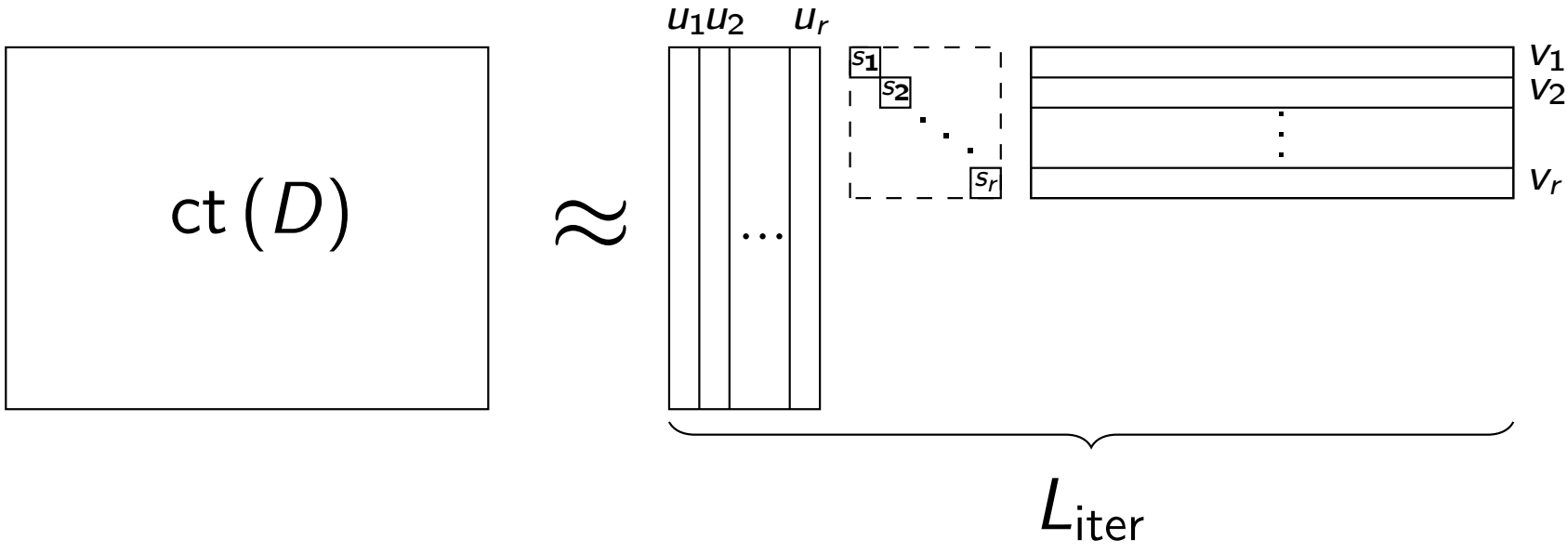
Results

Metric: Root mean square error (RMSE) with ground truth

Test RMSE for mean of each movie: 0.97

Test RMSE for rank-3 model: 0.89

Components?



u_1

$L[\text{movie}, \text{user}] := m_{\text{movie}} + \sum_{l=1}^{3} s_l u_l[\text{movie}] v_l[\text{user}]$

Highest entries:

1. Dante's Peak (0.29) FA score: 4.6/10
2. Volcano (0.28) FA score: 4.2/10
3. The Saint (0.24) FA score: 5.5/10

Lowest entries:

1. One Flew Over The Cuckoo's Nest (-0.10) 5 Oscars
2. The Godfather (-0.12) 3 Oscars
3. Casablanca (-0.13) 3 Oscars

u_2

$L[\text{movie}, \text{user}] := m_{\text{movie}} + \sum_{l=1}^{3} s_l u_l[\text{movie}] v_l[\text{user}]$

Highest entries:

1. Volcano (0.07)
2. Evita (0.05)
3. Dante's Peak (0.04)

Lowest entries:

1. Titanic (-0.16)
2. Star Wars (-0.16)
3. Raiders of the Lost Ark (-0.17)

u_3

$L[\text{movie}, \text{user}] := m_{\text{movie}} + \sum_{l=1}^{3} s_l u_l[\text{movie}] v_l[\text{user}]$

Highest entries:

1. Return of the Jedi (0.14)
2. Star Wars (0.13)
3. Raiders of the Lost Ark (0.13)

Lowest entries:

1. Leaving Las Vegas (-0.28)
2. Trainspotting (-0.29)
3. A Clockwork Orange (-0.30)

What have we learned?

How to perform collaborative filtering with low-rank models

How to fit these models to data